EZ204-DP PROFIBUS-DP Slave Interface

User Manual

May 2005

Warning!
Dangerous electrical voltage!

Before commencing the installation

- Disconnect the power supply of the device.
- Ensure that devices cannot be accidentally restarted.
- Verify isolation from the supply.
- Short circuit to earth.
- Cover or enclose neighboring units that are live.
- Follow the engineering instructions (AWA) of the device concerned.
- Only suitably qualified personnel in accordance with EN 50110-1/-2 (VDE 0105 Part 100) may work on this device/system.
- Before installation and before touching the device ensure that you are free of electrostatic charge.
- The functional earth (FE) must be connected to the protective earth (PE) or to the potential equalization. The system installer is responsible for implementing this connection.
- Connecting cables and signal lines should be installed so that inductive or capacitive interference does not impair the automation functions.
- Install automation devices and related operating elements in such a way that they are well protected against unintentional operation.
- Suitable safety hardware and software measures should be implemented for the I/O interface so that a line or wire breakage on the signal side does not result in undefined states in the automation devices.
- Ensure a reliable electrical isolation of the low voltage for the 24 volt supply. Only use power supply units complying with IEC 60364-4-41 (VDE 0100 Part 410) or HD 384.4.41 S2.
- Deviations of the mains voltage from the rated value must not exceed the tolerance limits given in the specifications, otherwise this may cause malfunction and dangerous operation.

- Emergency stop devices complying with IEC/EN 60204-1 must be effective in all operating modes of the automation devices.Unlatching the emergency-stop devices must not cause restart.

- Devices that are designed for mounting in housings or control cabinets must only be operated and controlled after they have been installed with the housing closed. Desktop or portable units must only be operated and controlled in enclosed housings.

- Measures should be taken to ensure the proper restart of programs interrupted after a voltage dip or failure. This should not cause dangerous operating states even for a short time. If necessary, emergency-stop devices should be implemented.

- Wherever faults in the automation system may cause damage to persons or property, external measures must be implemented to ensure a safe operating state in the event of a fault or malfunction (for example, by means of separate limit switches, mechanical interlocks etc.).

Contents

About This Manual

Target readership

This manual has been produced for automation technicians and engineers. A thorough knowledge of the PROFIBUS-DP fieldbus and the programming of a PROFIBUS-DP master is required. You should also be familiar with the operation of the EZ control relay or EZD multi-function display.

Other manuals on the device

The following operating manuals should be followed:

- EZ500/700 Series (MN05013003E)
- EZ800 Series (MN05013004E)
- EZD Series (MN05013005E)

All manuals are available on the Internet for download as PDF files. For a fast search enter the documentation number as the search criterion at www. EatonElectrical.com.

Device designation

This manual uses the following short names for equipment types, as far as the description applies to all of these types:

- EZ512-..-..., EZ7..-..-...

Type designation of the control relay, the point represents a placeholder for all characters used.

- EZ500 for
  - EZ512-AB...
  - EZ512-AC
  - EZ521-DA…
  - EZ512-DC
- EZ700 for
  - EZ719-AB…
  - EZ719-AC…
  - EZ719-DA…
  - EZ719-DC…
  - EZ721-DC…
- EZ800 for
  - EZ819-...
  - EZ820-...
  - EZ821-...
  - EZ822-...
- EZD-CP8.. for
  - EZD-CP8-ME
  - EZD-CP8-NT

- EZ-AB for
  - EZ512-AB...
  - EZ719-AB...
- EZ-AC for
  - EZ512-AC-..
  - EZ719-AC
  - EZ8..-AC-...
- EZ-DC for
  - EZ512-DC-..
  - EZ719-DC-..
  - EZ8..-.DC-...
- EZ-DA for
  - EZ512-DA...
  - EZ719-DA...

Abbreviations

This manual uses abbreviations with the following meanings:

hex:	Hexadecimal (number system with base 16)
dec :	Decimal (number system with base 10)
bcd:	binary coded decimal code
VR:	Value Range
PC:	Personal Computer

Writing conventions

Except for the first page of chapters and empty pages at the end, the top left of the page shows the chapter title and the top right of the page shows the current section for greater clarity.

▶indicates actions to be taken.

▽ **Attention!**
Warns of a hazardous situation that could result in damage to the product or components.

⚠ **Caution!**
Warns of the possibility of serious damage and slight injury.

⚡ **Warning**
Warns of the possibility of a hazardous situation that could result in major damage and serious or fatal injury or even death.

→ Indicates interesting tips and additional information

1 EZ204-DP

The EZ204-DP communication module was developed for automation tasks using the PROFIBUS-DP fieldbus. The EZ204-DP is a gateway and can only be used in conjunction with the EZ700, EZ800 or EZD basic units. The EZ control relay or EZD device with a PROFIBUS-DP gateway always works as a slave station on the network.

System overview

The EZ204-DP slaves are integrated in a PROFIBUS-DP system.

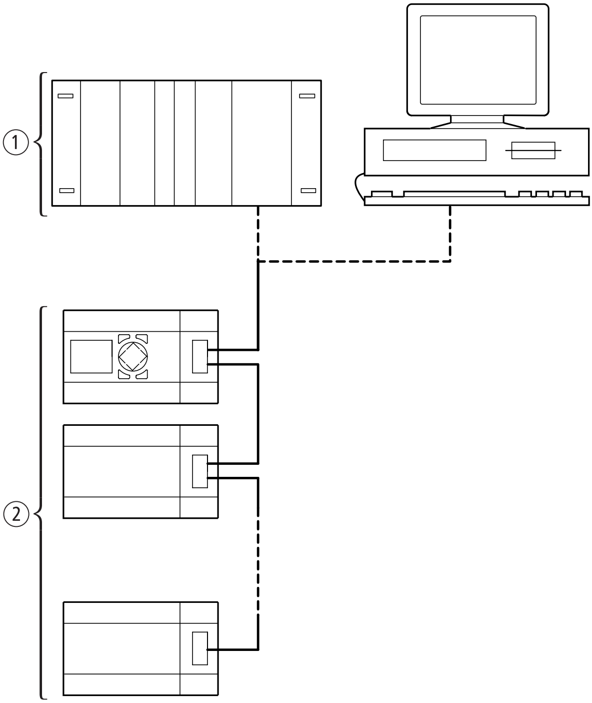


Figure 1: Integration of EZ204-DP in the DP network

① Master area, PLC or PC

② Slave area, e.g. EZ /EZD with DP interface

Device setup

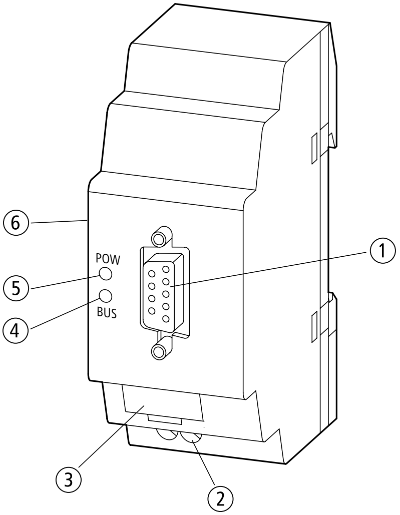


Figure 2: View of the device

① PROFIBUS-DP connection, 9-pole SUB-D socket

② 24 V DC supply voltage

③ Device designation plate

④ BUS communication LED

⑤ POW operation LED

⑥ EZ-LINK socket

Device function description

The EZ204-DP module allows the EZ and EZD series devices to be connected to a PROFIBUS-DP communication network. The following data can be transferred by selecting the appropriate SDO/PDO:

EZ700/800, EZD-CP8..

- S1 to S8
  Output data of the basic unit, RUN/STOP
  (read, as viewed from PROFIBUS-DP master)
- R1 to R16
  Input data of the basic unit, RUN/STOP
  (write, as viewed from PROFIBUS-DP master)
- All function relay data
  (read, as viewed from the PROFIBUS-DP master)
  - Timing relays
  - Counter relays
  - Time switches
  - Analog comparators
  - Weekday, time, summer/winter time (DST)
- The setpoints of the function relays
  (write, as viewed from PROFIBUS-DP master)
  - Timing relays
  - Counter relays
  - Time switches
  - Analog comparators
  - Weekday, time, summer/winter time (DST)

EZ800/EZD-CP8..

- All markers and EZ-NETdata
- Function blocks
  (read/write, as viewed from the master)
  - Arithmetic function blocks
  - Frequency counters, high-speed counters, incremental encoder counters
  - 7-day and year time switch
  - Operating hours counter
  - PID controllers
  - PWM (pulse width modulation)
  - Real-time clock

Improper use

EZ/EZD must not be used as a replacement for safety PLCs such as

- Burner,
- Emergency-stop,
- Crane or
- Two-hand safety controls.

2 Installation

The same principles apply as for EZ700, EZ800 and EZD basic units with expansion devices.

Connecting EZ204-DP to the basic unit

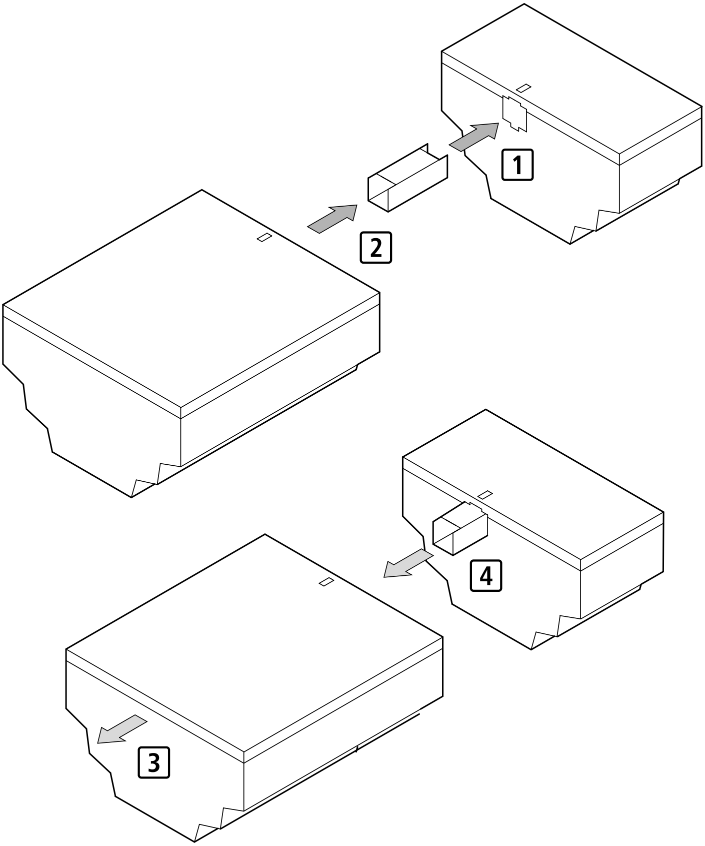


Figure 3: Fitting 1 + 2 or removing 3 + 4 the EZ204-DP to the basic unit

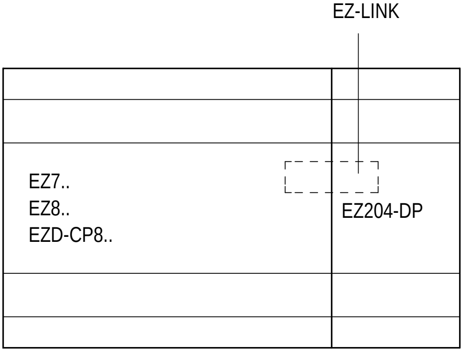


Figure 4: Connection between basic unit and EZ204-DP

Connecting the power supply

The EZ204-DP unit is run on a 24 V DC power supply (→ Section “Technical Data” from Page 187).

> **Warning**
> Always ensure electrical safety isolation between the extra low voltage (SELV) and the 24 V power supply.

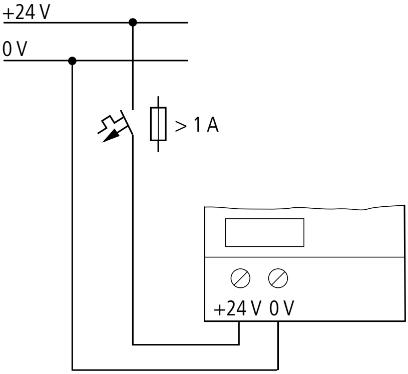
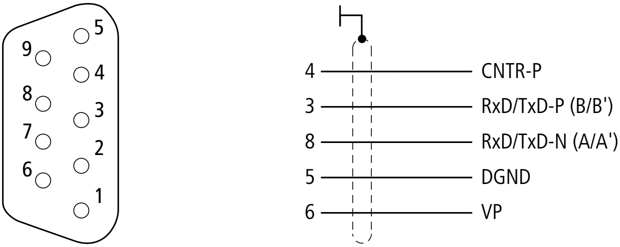


Figure 5: Standard connection

Connecting PROFIBUS-DP

Use a 9-pole SUB-D plug to connect the PROFIBUS-DP interface to the PROFIBUS-DP fieldbus. For this use the special PROFIBUS-DP plug and the special PROFIBUS-DP cable available from the Eaton range of accessories. The type of cable used determines the permissible maximum bus length and the transfer rate.

PROFIBUS-DP connection assignment



Pin	Signal name	Designation
1	Not assigned	-
2	Not assigned	-
3	RxD/TxD-P (B Line)	Receive/Send data P
4	CNTR-P / RTS	Request to Send
5	DGND	Data reference potential
6	VP	+5V DC for external bus connection
7	Not assigned	-
8	RxD/TxD-N (A-Line)	Receive/Send data N
9	Not assigned	-

Connections 3, 8 and the shield are sufficient for data transfer.

Bus terminating resistors

The first and last station in a bus segment must be connected to the bus with the bus terminating resistor switched on. The bus terminating resistor is switched externally. This external switch function can either be implemented as a separate bus terminating resistor or with a special Sub-D plug with an integrated bus termination.

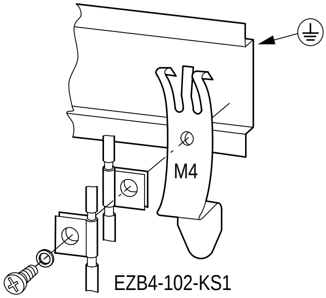
Eaton's PROFIBUS-DP data plug enables both bus terminating resistors to be switched on and off.

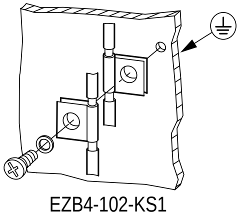
EMC wiring

Electromagnetic interference may have adverse effects on the communication fieldbus. This can be minimized in advance by taking suitable EMC measures. These include:

- System design in accordance with EMC requirements,
- EMC cabling and
- Measures that prevent the occurrence of large potential differences,
- Correct installation of the PROFIBUS system (cable, connection of bus connector,...).

The effects of electromagnetic interference can be significantly reduced by fitting the shield. The following two figures illustrate how to fit the shield.





Electrical isolation

The following electrical isolation should be provided for the interfaces of the EZ204-DP:

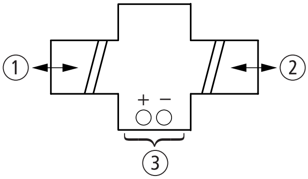


Figure 6: Potential isolation between the power supply and outputs

① Safe isolation of EZ-LINK 240 V AC

② Simple isolation of PROFIBUS-DP

③ 24 V DC supply voltage

Transfer rates – automatic baud rate detection

The EZ204-DP module automatically detects the baud rate used in the communication network after it is switched on. However, this requires that at least one station sends valid telegrams in the network.

The EZ204-DP module detects the transfer rate automatically. The following transfer rates are supported:

- 9.6 Kbit/s to 12000 Kbit/s

Maximum distances/bus cable lengths

Two types of bus cable are specified in IEC 61158. Cable type B should no longer be used with new applications because it is obsolete. Cable type A allows all transfer rates up to 12000 Kbit/s to be used. Cables for burial in the ground, festoon suspension and drum cables are also available.

The cable parameters are as follows:

Parameters	Cable type A
Surge impedance in Ω	135 ... 165 at 3 ... 20 MHz
Effective capacitance (pF/m)	< 30
Loop resistance (Ω/km)	< 110
Core diameter (mm)	> 0.64
Core cross-section (mm^2)	> 0.34

The cable parameters specified allow the following bus segment lengths.

Distance between stations when using Type A cable to IEC 61158:

Baud rate [Kbit/s]	Max. cable length Type A cable [m]
9.6	1200
19.2	1200
93.75	1200
187.5	1000
500	400
1500	200
3000	100
6000	100
12000	100

Distance between two stations when using Type B cable to IEC 61158:

Baud rate [Kbit/s]	Max. cable length Type B cable [m]
9.6	1200
19.2	1200
93.75	1200
187.5	1000
500	400
1500	–

3 Device Operation

Initial power up

▶Before you switch on the device, verify that it is properly connected to the power supply, to the bus connector and to the basic unit.

▶Switch on the power supply to the basic unit and the PROFIBUS-DP expansion unit.

The Power LED of the EZ204-DP is lit. The BUS LED is off (no communication via PROFIBUS-DP).

The GW message (intelligent station connected) is displayed on the basic unit.

Basic unit	GW display
EZ700	Flashing
EZ800	Flashing
EZD-CP8..	Flashing

As soon as the device is integrated in the PROFIBUS-DP network, the BUS LED is continuously lit (“static”) and the GW message is statically displayed, also on devices with a flashing GW message.

→ Valid data is only transferred via PROFIBUS-DP to the basic unit if the GW is displayed statically.

If the PROFIBUS-DP unit is factory set, the station address of the PROFIBUS-DP station must be set.

Setting the PROFIBUS-DP station address

Every PROFIBUS-DP station requires an unambiguous address in the PROFIBUS-DP structure. There are two ways of setting the PROFIBUS-DP addresses on the EZ204-DP:

- Using the integrated display and keypad on the EZ or EZD basic unit
- Using EZSoft on the PC.

Address range: 001 to 126

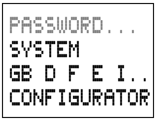
Setting the address on the basic unit with a display

Requirements:

- The basic unit (EZ700, EZ800 or EZD) and the EZ204-DP expansion unit must be fed with power.
- The basic unit has been unlocked (no password activated).
- The basic unit has a valid operating system version (→ page 15).
- The basic unit must be in STOP mode.
- The EZ204-DP is not communicating with the PROFIBUS-DP master (Bus LED is off).

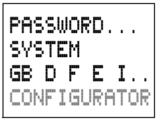
▶Enter the System menu by pressing DEL + ALT simultaneously.

DEL + ALT



```
PASSWORD...
SYSTEM
GB D F E I..
CONFIGURATOR
```

▶Use cursor buttons ⌃ or ⌄ to select CONFIGURATOR

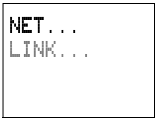


```
PASSWORD...
SYSTEM
GB D F E I..
CONFIGURATOR
```

▶Confirm your entry with OK

OK

▶With EZ800/EZD devices select the LINK...

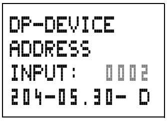


```
NET...
LINK...
```

▶Confirm with OK.

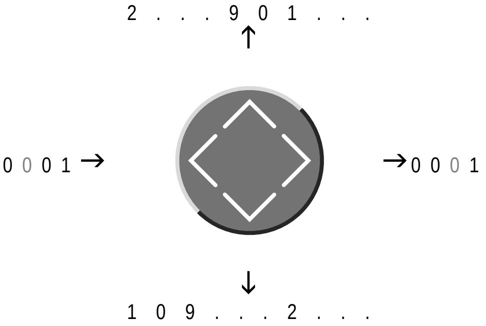
OK

EZ700 devices show the following dialog immediately:



```
DP-DEVICE
ADDRESS
INPUT:   0002
204-05.30- D
```

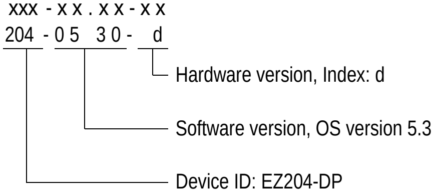
- ▶Set the address:
  Set the value of the current digit with ⌃ or ⌄ buttons.
  Move to the next digit with ‹ or ›.



▶Press OK to accept the address or

▶Abort address entry.

Information on the 4th display line:



Setting the address using EZSoft

Choose → Communication → Configuration → Expansion Devices →EZ204-DP.

→ The menu is only available in Communication View, therefore activate the Communication tab.

Status LEDs

The EZ204-DP expansion unit has two LEDs.

POW LED, Function

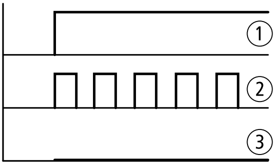


Figure 7: Function of the POW LED

① LED continuously lit:

– Power supply present

– Communication with the basic unit aborted

② LED flashing:

– Power supply present

– Communication with the basic unit correct

③ LED not lit:

– No power supply present

– Communication with the basic unit aborted

BUS LED, Function

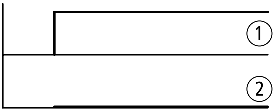


Figure 8: Function of the BUS LED

① LED continuously lit:

– PROFIBUS-DP communication correct

② LED not lit:

– No PROFIBUS-DP communication present

Cycle time of EZ basic unit

Communication between the basic unit and EZ204-DP via EZ-LINK increases the cycle time of the basic unit.

In extreme cases the cycle time may increase by 40 ms.

This should be taken into account for the reaction times of the basic unit.

4 PROFIBUS-DP Functions

Slave modules

The EZ204-DP expansion module is a PROFIBUS-DP slave in compliance with IEC 61186/EN 50170.

You can select the following EZ204-DP slave modules via the PROFIBUS-DP Configurator in the master PLC by using the appropriate GSD file. These modules are described in detail in chapters 5 and 6.

Module designation	Inputs (Bytes)	Outputs (Bytes)	Inputs/outputs (Bytes)	Service	Supported devices	Code in GSD file	Page
Control level							
1: Control commands, 9 bytes	–	–	9	• Real-time clock • Image • Function blocks	EZ700, EZ800, EZD-CP8..	0xB8	43
Input/output level							
2: Inputs, 3 bytes	3	–	–	• Read data: S1 – S8 • Operating mode	EZ700, EZ800, EZD-CP8..	0x92	36
3: Outputs, 3 bytes	–	3	–	• Write data: R1 – R8, R9 – R16 • Operating mode		0xA2	38
4: Inputs, 1 byte	1	–	–	• Read data: S1 – S8		0x90	38
5: Outputs, 1 byte	–	1	–	• Write data: R1 – R8, R9 – R16		0xA0	42

Diagnostics data

The EZ204-DP device features the standard diagnostics in accordance with the PROFIBUS specification.

Two additional diagnostics bytes are also sent.

Byte 0	Length of additional diagnostics bytes
Fixed 02_{hex}	00000010
Byte 1	Status of EZ-LINK
Value 00_{hex}	EZ-LINK is connected
Value 01_{hex}	EZ-LINK is disconnected

GSD file

A PROFIBUS-DP GSD file is required for selecting the device and for running it on the PROFIBUS-DP communication bus. The GSD file contains standard PROFIBUS station descriptions.

The file “EZ204DP.gsd” can be obtained at www.EatonElectrical.com. The file is also available on the EZSoft CD ROM.

PROFIBUS certification

EZ204-DP was certified as a PROFIBUS-DP device by the PROFIBUS User Organization. EZ204-DP contains the PROFIBUS VPC3+ interface.

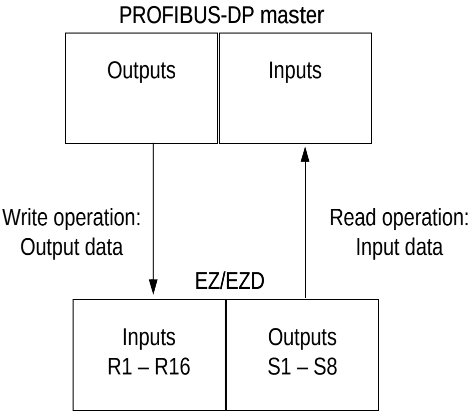
→ Irregular operation may occur under the following conditions:

- When in a multimaster system Class I and Class II DP masters with parameter or configuration data access the slave at the same time (highly unlikely).
- Or if other masters based on PROFIBUS layer 2 are connected.

5 Inputs/Outputs, EZ700/800/EZD Operating Mode

The appropriate module must be selected in the slave configuration in order for I/O data to be transferred between the EZ204-DP slave and a PROFIBUS-DP master.

→ The terms “input data” and “output data” are used from the point of view of the PROFIBUS-DP master.



"Inputs 3 bytes" module: operating mode, S1 – S8

The normal PROFIBUS-DP master data exchange with the EZ204-DP slave is via input data bytes 0, 1, 2.

Byte	Meaning	Value
0	Scan the operating mode	→ Table 1
1	Scan status of the EZ outputs S1 to S8	→ Table 2
2	Not assigned	00_{hex}

Requirement:
The "Inputs, 3 bytes" module must have been selected.

→ The output data and control commands can now only be used if you have selected the appropriate modules as well.

The master reads bytes 0, 1, 2 for the following data:

Table 1: Byte 0: Operating mode

EZ/EZD operating mode	Bit							
	7	6	5	4	3	2	1	0 Stop/Run
with debounce	0	0	0	1	0	0	0	0/1
without debounce	0	0	1	0	0	0	0	0/1

Example:

Value 21_{hex}	EZ/EZD is in Run mode and is working with input debounce.

Table 2: Byte 1: Status of S1 to S8 on the basic unit

Output	Bit							
	7	6	5	4	3	2	1	0
S1								0/1
S2							0/1	
S3						0/1		
S4					0/1			
S5				0/1				
S6			0/1					
S7		0/1						
S8	0/1							

Example:

Value 19_{hex}	S5, S4 and S1 are active.

Attention!
If control commands and I/O data are used at the same time:

- While the control command is being executed, the inputs will remain in the state before the control command was called.
- After the “Control commands” data exchange has been completed, the input bytes are refreshed.

"Inputs 1 byte" module: S1 – S8

When this module is selected, the master only receives 1 byte (coil output data S1 to S8) via PROFIBUS.

Byte	Meaning	Value
0	Scan status of the EZ outputs S1 to S8	→ Table 2 on Page 37

Requirement:
The "Inputs, 1 byte" module must have been selected.

→ The output data and control commands can now only be used if you have selected the appropriate modules as well.

"Outputs 3 bytes" module: operating mode, R9 – R16, R1 – R8

The normal PROFIBUS-DP master data exchange with the EZ204-DP slave is provided with output data bytes 0, 1, 2.

Byte	Meaning	Value
0	Set operating mode	→ Table 3
1	Set/reset the EZ/EZD inputs R9 to R16	→ Table 4
2	Set/reset the EZ/EZD inputs R1 to R8	→ Table 5

Requirement:
The "Outputs; 3 bytes" module must have been selected.

→ The output data and control commands can now only be used if you have selected the appropriate modules as well.

The master reads bytes 0, 1, 2 for the following data:

Table 3: Byte 0: Operating mode

EZ/EZD operating mode	Bit							
	7	6	5	4	3	2	1	0
Index for setting the basic unit to the safety state	0	0	0	0	0	0	0	0
Index for transferring valid data	0	0	0	1	0	1	0	0
RUN command	0	0	1	1	0	1	0	0
STOP command	0	1	0	0	0	1	0	0

0 = status “0”, 1 = status “1”

Explanation

Value 34_{hex} = $0011\,0100_{bin}$:
This value sets the EZ/EZD status from STOP to RUN. It is only interpreted as a command and therefore does not permit an additional transfer of data. The index value 14_{hex} must be used in this situation.

Value 44_{hex} = $0100\,0100_{bin}$:
This value sets the EZ/EZD status from RUN to STOP. It is also used only as command and it therefore works in the same way as the RUN command.

Table 4: Byte 1: Write status of R9 to R16

EZ/EZD Input	Bit							
	7	6	5	4	3	2	1	0
R9								0/1
R10							0/1	
R11						0/1		
R12					0/1			
R13				0/1				
R14			0/1					
R15		0/1						
R16	0/1							

Example:

Value 19_{hex} R13, R12 and R9 should be active.

Table 5: Byte 2: Write status of R1 to R8

EZ/EZD Input	Bit							
	7	6	5	4	3	2	1	0
R1								0/1
R2							0/1	
R3						0/1		
R4					0/1			
R5				0/1				
R6			0/1					
R7		0/1						
R8	0/1							

Example:

Value $2B_{hex}$ R6, R4, R2 and R1 should be active.

Attention!
If control commands and I/O data are used at the same time:

- While the control command is being executed, the inputs will remain in the state before the control command was called.
- After the “Control commands” data exchange has been completed, the output bytes are refreshed.

“Outputs 1 byte” module: R1 – R8

When this module is selected, the master only sends 1 byte (coil output data S1 to S8) via PROFIBUS.

Byte	Meaning	Value
0	Status of R1 to R8	→ Table 5 on Page 41

Requirement:
The “Outputs; 1 byte” module must have been selected.

→ The input data and control commands can now only be used if you have selected the appropriate modules as well.

Note on using the 1 byte modules

The 1 byte modules are not available in all device combinations. If any problems occur in handling, first check the state of the GW message in the status display of the basic unit:

GW static:	The 1-byte mode can be used
GW flashing:	Check the device version of the EZ204-DP and the basic unit. If these are valid, check the configuration in the PROFIBUS network and the Configurator.

6 Control commands for EZ700

Data exchange procedure

The “Control commands 9 bytes” module allows extended data exchange of the EZ700 on the PROFIBUS-DP communication bus. This allows you to transfer services from the following areas:

- Read/write image date and time (Page 45)
- Read/write image data (Page 49) and
- Read/write function block data (Page 72).

A data exchange procedure is required in order to ensure the safe exchange of data via PROFIBUS-DP from master to slave and vice versa.

Attention!

While a control command is being executed, the input and output data will remain in the state before the control command was called. Only after the “Control commands” data exchange has been completed, will the I/O data be refreshed.

Caution!
Only those values specified for the command code should be used.

Check the values that you write in order to avoid malfunctions.

Requirement:
The “Control commands 9 byte” module must have been selected.

The master initiates the data exchange of the control commands and the addressed slave responds.

During communication 9 data bytes (byte 0 = toggle byte, bytes 1 to 8 information bytes) are sent via PROFIBUS.

The basic telegram structure is shown in the following diagram.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8

Byte 0 – Toggle byte

Byte 0 is used to activate the sending of a control command with the toggle function.

Bit	7	6	5	4	3	2	1	0
01_{hex}/861_{hex}	0/1	0	0	0	0	0	0	1
	Toggle bit	fixed						

Procedure

▶ To send a command, bit 7 must be toggled, i.e. set either from 1 to 0 or from 0 to 1.

▶ Then poll the toggle bit for the coupling modules response until it has the same status as the toggle bit sent. This status indicates to the master that the response to the sent command is valid.

▶ Do not send a new command until you have received a response (changing of the toggle bit), otherwise the response of the previous command will be overwritten before it can be read.

→ In order to use input/output data and control commands simultaneously:

Only after the “Control commands” data exchange has been completed, will the I/O data be refreshed.

All specified commands and parameters must be transferred in hexadecimal format.

The following tables show the different control commands possible. These essential control commands fall into three essential categories – real-time clock, image and function blocks.

→ Please also note the relevant description of the real-time clock provided in the EZ500/700 manual (MN05013003E).

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command		
	Read	93	–
	Write	B3	–
	Response		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Len	05	05
3	Index	0 – 2[1]	0 – 2[1]
4 – 8	Data 1 – 4	depending on index, → Table 6	

[1] 0 = Time/date, → Table 6
1 = Summer time, → Table 7
2 = Winter time, → Table 8

Table 6: Index 0 – date and time of real-time clock

Byte	Contents	Operand		Value (hex)
4	Data 1	Hour	0 to 23	0x00 to 0x17h
5	Data 2	Minute	0 to 59	0x00 to 0x3Bh
6	Data 3	Day	Day (1 to 28; 29, 30, 31; depending on month and year)	0x01 to 0x1Fh
7	Data 4	Month	1 to 12	0x01 to 0x0Ch
8	Data 5	Year	0 to 99 (corresponds to 2000-2099)	0x00 to 0x63h

Table 7: Index 1 – Summer time

Byte	Contents		Value (hex)
4	Data 1	Area	
		None	00
		Rule	01
		Automatic EU	02
		Automatic GB	03
		Automatic US	04
for “Area” = “Rule”:			
5	Data 2	Summer time switching rule	→ Table 9
6	Data 3		
7	Data 4		
8	Data 5		

Table 8: Index 2 – Winter time
(only valid if Area = “Rule” selected)

Byte	Contents		Value (hex)
4	Data 1	Area = Rule	01
5 – 8	Data 2 – 5	Winter time switching rule	→ Table 9

Switching rule bit array

→ Please also read the detailed description in the EZ500/700 manual (MN05013003E).

The following table shows the composition of the corresponding data bytes.

Table 9: Switching rule bit array

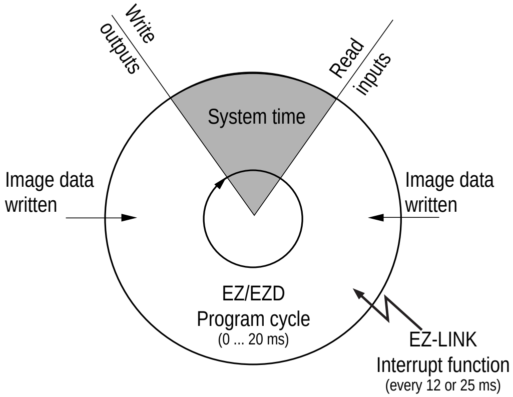
	Data 5		Data 4	Data 3		Data 2		
Bit	31 30 29 28	27 26 25 24 23 22	21 20 19 18 17	16 15 14 13	12 11 10 9 8	7 6	5 4 3	2 1 0
	Difference	Time of time change		Month	Day	Rule_2	Day	Rule_1
	0: 0:30h	Minute: 0 to 59	Hour: 0 to 23	0 to 11	0 to 30	0: month	0: Su	0: on
	1: 1:00h					1: after the	1: Mo	1: on the first
	2: 1:30h					2: before the	2: Tu	2: on the second
	3: 2:00h						3: We	3: on the third
	4: 2:30h						4: Thu	4: on the fourth
	5: 3:00h						5: Fr	5: on the last
							6: Sa	

Read/write image data

→ Please also observe the relevant description of possible image data provided in the EZ500/700 manual (MN05013003E) or in the EZSoft Help.

The latest edition of the manual is available as a PDF file from the Internet at: www.EatonElectrical.com. Search Term: MN05013003E.

General notes on working with image data



When writing to image data, it must be remembered that an image (e.g. inputs, outputs,...) used in the EZ700/EZ800/EZD program is also written cyclically by the actual program. The only image data that is unchanged is the data that is not used in the program and is therefore not overwritten in the program cycle. This operating principle also means that an image written via EZ-LINK, such as output data is only then output at the physical outputs of the EZ700/EZ800/EZD when the control relay is in Run mode.

Overview

Operands	Meaning	Read/ write	Type	Page
A1 – A16	Analog value comparators/threshold comparators: A1 – A16	Read	8C	51
C1 – C16	Counters: C1 – C16	Read	EE	52
D1 – D16	Text function blocks: D1 – D16	Read	94	53
I1 – I16	Local inputs: I1 – I16	Read	84	54
IA1 – IA4	Local analog inputs: IA1 – IA4	Read	8C	56
M1 – M16, N1 – N16	Markers: M1 – M16/N1 – N16	Write	86/87	58
M1 – M16, N1 – N16	Markers: M1 – M16/N1 – N16	Read	86/87	60
O1 – O4	Operating hours counters: O1 – O4	Read	EF	62
P1 – P4	Local P buttons: P1 – P4	Read	8A	63
Q1 – Q8	Local outputs: Q1 – Q8	Read	85	65
R1 – R16/ S1 – S8	Inputs/outputs of EZ-LINK: R1 – R16/S1 – S8	Read	88/89	66
T1 – T16	Timing relays: T1 – T16	Read	ED	68
Y1 – Y4	Year time switch: Y1 – Y8	Read	91	69
Z1 – Z3	Master reset: Z1 – Z3	Read	93	70
H1 – H4	7-day time switch: ⌚1 – ⌚8	Read	90	71

Analog value comparators/threshold comparators: A1 – A16

The following commands are used to read the logic state of the individual analog value comparators A1 to A16.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	8C	8C
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 10
6	Data 2 (Low Byte)	00	→ Table 10
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

Table 10: Byte 5 to 6: Data 1 to 2

Data 1	Bit	7	6	5	4	3	2	1	0
A1									0/1
A2								0/1	
...					...				
A8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
A9									0/1
A10								0/1	
...					...				
A16		0/1							

Counters: C1 – C16

The following commands are used to read the logic state of the individual counters C1 – C16.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	EE	EE
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 20
6	Data 2 (Low Byte)	00	→ Table 20
7 – 8	Data 3 – 4	00	00

1) Possible causes → page 91

Table 11: Byte 5 to 6: Data 1 to 2

Data 1	Bit	7	6	5	4	3	2	1	0
C1									0/1
C2								0/1	
...					...				
C8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C9									0/1
C10								0/1	
...					...				
C16		0/1							

Text function blocks: D1 – D16

The following commands are used to read the logic state of the individual text function blocks (D markers).

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	94	94
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 12
6	Data 2 (High Byte)	00	→ Table 12
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

Table 12: Byte 5 to 6: Data 1 to 2

Data 1	Bit 7	6	5	4	3	2	1	0
D1								0/1
D2							0/1	
...				...				
D8	0/1							
Data 2	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D9								0/1
D10							0/1	
...				...				
D16	0/1							

Local inputs: I1 – I16

This command string enables you to read the local inputs of the EZ700 basic unit. The relevant input word is stored in Intel format.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	02	02
3	Type	84	84
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 13
6	Data 2 (High Byte)	00	→ Table 13
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

Table 13: Byte 5 to 6: Data 1 to 2

Data 1	Bit	7	6	5	4	3	2	1	0
I1									0/1
I2								0/1	
..					..				
I8		0/1							
Data 2	Bit	7	6	5	4	3	2	1	0
I9									0/1
I10								0/1	
..					..				
I16		0/1							

Local analog inputs: IA1 – IA4

The analog inputs on the EZ700 basic unit (I7, I8, I11, I12) can be read directly via PROFIBUS-DP. The 16-bit value is transferred in Intel format (Low Byte first).

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	02	02
3	Type	8C	8C
4	Index	00 – 03[2]	00 – 03[2]
5	Data 1 (Low Byte)	00	→ Table 14
6	Data 2 (High Byte)	00	→ Table 14
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

[2] 00 = Analog input I7
01 = Analog input I8
02 = Analog input I11
03 = Analog input I12

Example:
A voltage signal is present at analog input 1. The required telegrams for reading the analog value are as follows:

Table 14: Example telegram for reading the value at the analog input

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response: Read successful	–	C2
2	Len	02	02
3	Type	8C	8C
4	Index	02[1]	02[1]
5	Data 1	00	4B
6	Data 2	00	03
7	Data 3	00	00
8	Data 4	00	00

[1] 02 = Analog input I11

Byte 5 – Data 1 (Low Byte): $4B_{hex}$
Byte 6 – Data 2 (High Byte): 03_{hex}
→ corresponding 16-bit value: $034B_{hex} = 843$

The value 843 corresponds to the IO bit value of the analog converter. The following conversion is required for the actual analog value:

$$\frac{10\ \text{V}}{1023} \times \text{IO bit value} \quad => \quad \frac{10\ \text{V}}{1023} \times 843 = 8.24\ \text{V}$$

Markers: M1 – M16/N1 – N16

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Write	8C	–
	Response:		
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Len	01	01
3	Type[2]		
	With M marker	86	86
	With N marker	87	87
4	Index[2]	00 – 0F	00 – 0F
5	Data 1 (Low Byte)[3]	00/01	00/01
6 – 8	Data 2 – 4	00	00

1) Possible causes → page 91

2) There are 16 M markers and 16 N markers. The markers are addressed by Type and Index:
Use Type to select the M or N marker.
Use Index to select the marker number.

3) The marker is set if a value is written to the data byte that does not equal zero. The marker is reset accordingly if the value 0 is written to data byte Data 1.

Example:
Marker M13 is set.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Write	8C	–
	Response:		
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Len	01	01
3	Type		
	M marker	86	86
4	Index	0C	0C
5	Data 1	01	00
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Markers: M1 – M16/N1 – N16

Unlike the write operation, the marker read operation reads the entire marker area of a particular marker type (M or N) is read.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type		
	M marker	86	86
	N marker	87	87
4	Index[2]	00	00
5	Data 1 (Low Byte)	00	→ Table 15
6	Data 2 (Low Byte)	00	→ Table 15
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

[2] There are 16 M markers and 16 N markers. The markers are addressed by Type and Index:
Use Type to select the M or N marker.
Use Index to select the marker number.

Table 15: Byte 5 to 6: Data 1 to 2

Data 1		Bit	7	6	5	4	3	2	1	0
M	N									
M1	N1									0/1
M2	N2								0/1	
...	...					...				
M8	N8		0/1							
Data 2		**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M9	N9									0/1
M10	N10								0/1	
...	–					...				
M16	N16		0/1							

Example:
The N markers are read:

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type		
	N marker	87	87
4	Index	00	00
5	Data 1 (Low Byte)	00	04
6	Data 2 (Low Byte)	00	84
7 – 8	Data 3 – 4	00	00

1) Possible causes → page 91

The markers N3, N11 and N16 are set.

Operating hours counters: O1 – O4

The following commands are used to read the logic state of the operating hours counters O1 – O4.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	EF	EF
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 16
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Table 16: Byte 5: Data 1

Data 1	Bit	7	6	5	4	3	2	1	0
O1									0/1
O2								0/1	
O3							0/1		
O4						0/1			
...		...	...	...	...				

Local P buttons: P1 – P4

The local P buttons are the display cursor buttons of the EZ700 basic unit. You can scan the buttons in both RUN and STOP mode.

→ Ensure that the P buttons are also activated via the System menu (in the basic unit).

Only one byte has to be transferred for the P buttons.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	8A	8A
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 17
6 – 8	Data 2 – 4	00	00

1) Possible causes → page 91

Table 17: Byte 5: Data 1

Data 1	Bit	7	6	5	4	3	2	1	0
P1									0/1
P2								0/1	
P3							0/1		
P4						0/1			
–					0				
–				0					
–			0						
–		0							

Example:
Data 1 = 2_{hex} → P3 is active.

Local outputs: Q1 – Q8

The local outputs can be read directly via the PROFIBUS-DP fieldbus.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	85	85
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 18
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Table 18: Byte 5: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
Q1								0/1
Q2							0/1	
..				..				
Q8	0/1							

Example:
Data 1 = 52_{hex} → Q2, Q5 and Q7 are active.

Inputs/outputs of EZ-LINK: R1 – R16/S1 – S8

This service allows you to read the local R and S data and the data of the NET stations (1 – 8) transferred via EZ-LINK, again from the relevant EZ700 image.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type		
	for R data	88	88
	for S data	89	89
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 19
6	Data 2 (Low Byte)	00	→ Table 19
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

Table 19: Byte 5 to 6: Data 1 to 2

Data 1		Bit	7	6	5	4	3	2	1	0
RW	SW									
R1	S1									0/1
R2	S2								0/1	
...	...					...				
R8	S8		0/1							
Data 2		Bit	7	6	5	4	3	2	1	0
R9	–									0/1
R10	–								0/1	
...	–					...				
R16	–		0/1							

Timing relays: T1 – T16

The following commands are used to read the logic state of the individual timers T1 - T16.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	ED	ED
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 20
6	Data 2 (Low Byte)	00	→ Table 20
7 – 8	Data 3 – 4	00	00

[1] Possible causes → page 91

Table 20: Byte 5 to 6: Data 1 to 2

Data 1	Bit	7	6	5	4	3	2	1	0
T1									0/1
T2								0/1	
...					...				
T8		0/1							
Data 2	**Bit**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T9									0/1
T10								0/1	
...					...				
T16		0/1							

Year time switch: Y1 – Y8

The following commands are used to read the logic state of the individual year time switches.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	91	91
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 21
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Table 21: Byte 5: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
HY1								0/1
HY2							0/1	
HY3						0/1		
HY4					0/1			
HY5				0				
HY6			0					
HY7		0						
HY8	0							

Example:
Data 1 = 1_{hex} l HY2 is active

Master reset: Z1 – Z3

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	93	93
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 22
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Table 22: Byte 5: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
Z1 for Q outputs								0/1
Z2 for M markers							0/1	
Z3 for outputs and markers						0/1		
...	0	0	0	0	0			

7-day time switch: Ö1 – Ö8

The following commands are used to read the logic state of the individual 7-day time switches.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command: Read	88	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0[1]
2	Len	01	01
3	Type	90	90
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 23
6 – 8	Data 2 – 4	00	00

[1] Possible causes → page 91

Table 23: Byte 5: Data 1

Data 1	Bit 7	6	5	4	3	2	1	0
HW1								0/1
HW2							0/1	
HW3						0/1		
HW4					0/1			
HW5				0				
HW6			0					
HW7		0						
HW8	0							

Example:
Data 1 = 2_{hex} l Ö3 is active.

Read/write function block data

> → Please also observe the relevant description of the function blocks provided in the EZ500/700 manual (MN05013003E) or in the EZSoft Help.

General notes

Always note the following when working with function blocks:

- The relevant data is transferred in Intel format. In other words, the first byte is the low byte (Byte 5) and the last byte (byte 8) the high byte.
- The maximum data length is 4 bytes. All values must be transferred in hexadecimal format.

Overview

Operands	Meaning	Read/ write	Type	Page
A1 – A16	Analog value comparator/threshold comparator: A1 – A16	Read/write	8D	73
C1 – C16	Counter relays: C1 – C16	Read/write	8F	76
O1 – O4	Operating hours counters: O1 – O4	Read/write	92	79
T1 – T16	Timing relays: T1 – T16	Read/write	8E	81
Y1 – Y8	Year time switch: Y1 – Y8	Read/write	A2	85
⌚1 – ⌚8	7-day time switch: ⌚1 – ⌚8	Read/write	A1	88

Analog value comparator/threshold comparator: A1 – A16

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	8D	8D
3	Instance[2]	00 – 0F	00 – 0F
4	Index	Table 24	
5 – 8	Data 1 – 4	depending on index → Table 25	

1) Possible causes → page 91

2) EZ provides 16 analog comparators A1 to A16 for use as required. These can be addressed using the instance (0 – F).

Table 24: Operand overview

Index (hex)	Operand		Read	Write
00	Parameters → Table 25		×	
01	Control byte → Table 26		×	
02	Comparison value 1	I1[2]	×	c[1]
03	Comparison value 2	I2[2]	×	c[1]
04	Gain factor for I1 (I1 = F1 × I1)	F1[2]	×	c[1]
05	Gain factor for I2 (I2 = F2 × I2)	F2[2]	×	c[1]
06	Offset for value I1 (I1 = OS + actual value at I1)	OS[2]	×	c[1]
07	Switching hysteresis for value I2	HY[2]	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

[2] A 16-bit value is transferred in data bytes Data 1 – Data 2. It should be remembered that the low byte 1 is in Data 1 (Byte 5) and the high byte 2 (byte 8) in Data 2.
Example: 5327_{dec} = $14CF_{hex}$ l Data 1 = 0 × CF, Data 2 = 0 × 14

Table 25: Index 00 – Parameters

Meaning	Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Appears in the parameter menu																	
Yes/no																	0/1
Compare																	
FB not used														0	0	0	
EQ (=)														0	0	1	
GE (≧)														0	1	0	
LE (≦)														0	1	1	
GT (>)														1	0	0	
LT (<)														1	0	1	
Use as constant and therefore can be written to																	
I1= Constant													0/1				
F1= Constant												0/1					
I2= Constant											0/1						
F2 = Constant										0/1							
OS = Constant									0/1								
HY = Constant								0/1									
Not used		0	0	0	0	0	0										

Example:
Data 1 (Byte 5) = 0xA3, Data 2 (Byte 6) = 0x03
l Resulting 16-bit value = $03A3_{hex}$

Meaning: HY, OS, F2, F1 are assigned a constant; I1, I2 are assigned to a variable such as I7, I8 C2...etc., appears in the Parameter menu;

The output of the analog value comparator is active for as long as the comparison (I1 × F1) + OS = (I2 × F2) + HY is fulfilled.

Table 26: Index 01 – Control byte

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	Q1[1]

1) Status 1 if comparison condition is fulfilled.

Counter relays: C1 – C16

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	8F	8F
3	Instance[2]	00 – 0F	00 – 0F
4	Index	→ Table 27	
5 – 8	Data 1 – 4	depending on index,→ Table 28	

1) Possible causes → page 91

2) EZ provides 16 counters C1 to C16 for use as required. These can be addressed using the instance (0 – F).

Table 27: Operand overview

Index (hex)	Operand		Read	Write
00	Parameters → Table 28		×	
01	Control byte → Table 29		×	
02	Actual value	S1[2]	×	c[1]
03	Counter setpoint 2	S2[2]	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

[2] A 16-bit value is transferred in data bytes Data 1 – Data 2. It should be remembered that Data 1 is the low byte and Data 2 the high byte.

Table 28: Index 00 – Parameters

Meaning	Bit	7	6	5	4	3	2	1	0
Appears in the parameter menu									
Yes/no									0/1
Counter mode									
FB not used							0	0	
Up/down counter (N)							0	1	
High-speed up/down counter (H)							1	0	
Frequency counter (F)							1	1	
Use as constant and therefore can be written to									
Counter setpoint S1						0/1			
Unused bits		–	–	–	–				

Example:
Data 1 (Byte 5) = 0x07

Meaning:
The values appear in the Parameter menu. The counter is used in the mode of the frequency meter. The counter setpoint 1 is not assigned to a constant and cannot therefore be written to.

Table 29: Index 01 – Control byte

Data 1	Bit	7	6	5	4	3	2	1	0
FB output		–	–	–	–	C[4]	RE[3]	D[2]	Q1[1]

1) Switch contact
2) Count direction: 0 = up counting,
   1 = down counting
3) Reset, the timing relay is reset (reset coil)
4) Count coil, counts on every rising edge

Example:
the actual value of C3 is to be read:

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Command: Read	89	–
	Response: Read successful	–	C2
1	Type	8F	8F
2	Instance	02	02
3	Index	02	02
4	Data1	00	12
5	Data 2	00	03
6	Data 3	00	00
7	Data 4	00	00

Explanation:

Data 1 = 12
Data 2 = 03
l resulting 16-bit value = 0312_{hex} = 786_{dec}

Counter status = 786

Operating hours counters: O1 – O4

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	92	92
3	Instance[2]	00 – 03	00 – 03
4	Index	→ Table 30	
5 – 8	Data 1 – 4	depending on index, → Table 31	

[1] Possible causes → page 91

[2] EZ provides 4 operating hours counters O1 to O4. These can be addressed using the instance (0 – 3).

Table 30: Operand overview

Index (hex)	Operand		Read	Write
00	Parameters → Table 31		×	
01	Control byte → Table 32		×	
02	Actual value	S1[2]	×	c[1]
03	Counter setpoint 2	S2[2]	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

[2] A 32-bit value is transferred in data bytes Data 1 – Data 4. It should be remembered that the Data 1 is the low byte and Data 4 the high byte.

Table 31: Index 00 – Parameters

Meaning	Bit	7	6	5	4	3	2	1	0
Appears in the parameter menu									
Yes/no									0/1
Use in the program									
Setpoint S1								0/1	
Unused bits		–	–	–	–	–	–		

Example:
Data 1 (Byte 5) = 0x01

Meaning:
The values appear in the Parameter menu.

Table 32: Index 01 – Control byte

Data 1	Bit	7	6	5	4	3	2	1	0
FB output		–	–	–	–	–	RE[3]	EN[2]	Q1[1]

1) Switch contact
2) Enable, the timing relay is started (trigger coil)
3) Reset, the timing relay is reset (reset coil)

Example:
Index 02/03

Transferred values:	Data 1 0x21 Data 2 0x23 Data 3 0x40 Data 4 0x00
Resulting value:	$00402321_{hex} = 4203297_{dec}$

Timing relays: T1 – T16

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	8E	8E
3	Instance[2]	00 – 0F	00 – 0F
4	Index	→ Table 33	
5 – 8	Data 1 – 4	depending on index → Table 34	

[1] Possible causes → page 91

[2] EZ provides 16 timing relays T1 to T16 for use as required. These can be addressed using the instance (0 – F).

Table 33: Operand overview

Index (hex)	Operand		Read	Write
00	Parameters → Table 34		×	
01	Control byte → Table 35		×	
02	Actual value 1	T	×	c[1]
03	Time setpoint 1	S1[2]	×	c[1]
04	Time setpoint 2	S2[2]	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

[2] A 16-bit value is transferred in data bytes Data 1 – Data 2. It should be remembered that Data 1 is the low byte and Data 2 the high byte.

Table 34: Index 00 – Parameters

Meaning	Bit	7	6	5	4	3	2	1	0
Appears in the parameter menu									
Yes/no									0/1
Timer mode									
On-delayed						0	0	0	
Off-delayed						0	0	1	
On-delayed with random setpoint						0	1	0	
Off-delayed with random setpoint						0	1	1	
On and off delayed (two time setpoints)						1	0	0	
On and off delayed each with random setpoint (two time setpoints)						1	0	1	
Impulse transmitter						1	1	0	
Flashing relay (two time setpoints)						1	1	1	
Time base									
FB not used				0	0				
Millisecond: S				0	1				
Second: M:S				1	0				
Minute: H:M				1	1				
Use as constant and therefore can be written to									
Time setpoint S1			0/1						
Time setpoint S2		0/1							

Example:
Data 1 (Byte 5) = 0xAC

Meaning:
The values appear in the Parameter menu. The time is used in the impulse transmitter mode with the Second time base. The time setpoint S1 is assigned a constant and the time setpoint S2 is assigned a variable such as I7, I8 C2...etc.

Table 35: Index 01 – Control byte

	Bit	7	6	5	4	3	2	1	0
FB input/output Data 3		–	–	–	–	ST[4]	RE[3]	EN[2]	Q1[1]

1) Switch contact
2) Enable, the timing relay is started (trigger coil)
3) Reset, the timing relay is reset (reset coil)
4) Stop, the timing relay is stopped (Stop coil)

Example:
The time setpoint 1 is to be read:

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Command: Read	89	–
	Response: Read successful	–	C2
1	Type	8E	8E
2	Instance	00	00
3	Index	03	03
4	Data1	00	4C
5	Data 2	00	06
6	Data 3	00	00
7	Data 4	00	00

Explanation:

Data 1 = 4C
Data 2 = 06
→ resulting 16-bit value = $064C_{hex}$ = 1612_{dec}

Meaning depending on set time base:

Millisecond	S	16120 ms	16.120 s
Seconds	M:S	1620 s	26:52 Minutes
Minute	H:M	1612 min	67:04 Hours

Year time switch: Y1 – Y8

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	A2	A2
3	Instance[2]	00 – 07	00 – 07
4	Index	→ Table 36	
5 – 8	Data 1 – 4	depending on index → Table 37	

[1] Possible causes → page 91

[2] EZ provides 8 year time switches Y1 to Y8 for use as required. These can be addressed using the instance (0 – 7).

Table 36: Operand overview

Index (hex)	Operand	Read	Write
00	Parameters→ Table 37	×	
01	Control byte→ Table 38	×	
	Channel A	×	c1
11	Time point ON	×	c1
12	Time point OFF	×	c1
	Channel B	×	c1
21	Time point ON	×	c1
22	Time point OFF	×	c1
	Channel C	×	c1
31	Time point ON	×	c1
32	Time point OFF	×	c1
	Channel D	×	c1
41	Time point ON	×	c1
42	Time point OFF	×	c1

1) The value can only be written if it is assigned to a constant in the program.

2) The switch points are transferred in data bytes Data 1 – Data 3.

Table 37: Index 00 – Parameters

Meaning	Bit	7	6	5	4	3	2	1	0
Appears in the parameter menu									
Channel A									0/1
Channel B								0/1	
Channel C							0/1		
Channel D						0/1			
Unused bits		–	–	–	–				

Example:
Data 1 (Byte 5) = 0x03 → The values of the year time switch of channel A and B in the parameter menu.

Table 38: Index 01 – Control byte

Data 1	Bit	7	6	5	4	3	2	1	0
FB output		–	–	–	–	–	–	–	Q1[1]

[1] Status 1, if the count condition is fulfilled.

Channel A, Index 11/12
Index 0x11 channel A ON time
Index 0x12 channel A OFF time
Data 1 (Byte 5) – Day
Data 2 (Byte 6) – Month
Data 3 (Byte 7) – Year

Example:
The year time switch channel A is to be activated on the 21.04.2004.

Index = 0x11
Data 1 = 0x15
Data 2 = 0x04
Data 3 = 0x04

The year time switch channel B is to be deactivated on the 05.11.2012.

Index = 0x22
Data 1 = 0x05
Data 2 = 0x0B
Data 3 = 0x0C

7-day time switch: ⏲1 – ⏲8

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 44	
1	Command:		
	Read	89	–
	Write	8D	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0[1]
2	Type	A1	A1
3	Instance[2]	00 – 07	00 – 07
4	Index	→ Table 39	→ Table 39
5 – 8	Data 1 – 4	depending on index, Table 40	

[1] Possible causes → page 91

[2] EZ provides 8 7-day time switches ⏲1 to ⏲8 for use as required. These can be addressed using the instance (0 – 7).

Table 39: Operand overview

Index (hex)	Operand		Read	Write
00	Parameters→ Table 40		×	
01	Control byte→ Table 41		×	
11	Channel A	Day on/off	×	c1
12		On time	×	c1
13		Off time	×	c1
21	Channel B	Day on/off	×	c1
22		On time	×	c1
23		Off time	×	c1
31	Channel C	Day on/off	×	c1
32		On time	×	c1
33		Off time	×	c1
41	Channel D	Day on/off	×	c1
42		On time	×	c1
43		Off time	×	c1

1) The value can only be written if it is assigned to a constant in the program.

2) A 16-bit value is transferred in data bytes Data 1 – Data 4. It should be remembered that Data 1 is the low byte and Data 2 the high byte.

Table 40: Index 00 – Parameters

Meaning	Bit	7	6	5	4	3	2	1	0
Appears in the parameter menu									
Channel A									0/1
Channel B								0/1	
Channel C							0/1		
Channel D						0/1			
Unused bits		–	–	–	–				

Example:
Data 1 (Byte 5) = 0x03

Meaning:
The values of the WH.. 7-day time switch from channel A and B appear in the parameter menu.

Table 41: Index 01 – Control byte

Data 1	Bit	7	6	5	4	3	2	1	0
FB output		–	–	–	–	–	–	–	Q1[1]

[1] Status 1, if the count condition is fulfilled.

Channel A, Index 11/12/13
Index 0x11 channel A Weekday on/off
Data 1 (Byte 5) – Weekday on
Data 2 (Byte 6) – Weekday off
0x01 = Sunday ... 0x07 = Saturday

The 16-bit value equals 0x00 if the channel is not used.

Index 0x12 – On time (2 Byte)
Index 0x13 – Off time (2 Byte)
Data 1 (Byte 5) – Hour
Data 2 (Byte 6) – Minute

Example: On time at 13:43 p.m.
Data 1 = 0x0D
Data 2 = 0x2B

Analysis – error codes via EZ-LINK

The EZ700 basic unit will return a defined error code in the event of an incorrectly selected operating mode or an invalid telegram. The error code transferred has the following structure:

Telegram structure

Byte	Meaning	Slave transmits (value hex)
0	Toggle byte	→ page 44
1	Response	
	Command rejected	C0
2	Type	00
3	Instance	00
4	Index	00
5	Error code	→ Table 42

Table 42: Error codes

Error code	Description
0x01	Unknown telegram transmitted.
0x02	Unknown object transmitted.
0x03	Unknown command transmitted.
0x04	Invalid instance transmitted.
0x05	Invalid parameter set transmitted.
0x06	An attempt was made to write to a variable that is not a constant.
0x0C	The device is in an invalid device mode. STOP l RUN or RUN l STOP
0x0D	Invalid display access. Exit the menu level so that the status display is showing in the display. The clock cannot be written to.
0xF0	Attempt made to control an unknown parameter.
0xF1	Impermissible value

7 EZ800/EZD Control Commands

Data exchange procedure

The Control commands 9 bytes module allows extended data exchange of the EZ800 and the EZD on the PROFIBUS-DP communication bus. This allows you to transfer services from the following areas:

- Read/write date and time (Page 95)
- Read/write image data (Page 99) and
- Read/write function block data (Page 120).

A data exchange procedure is required in order to ensure the safe exchange of data via PROFIBUS-DP from master to slave and vice versa.

▽ Attention!

While a control command is being executed, the input and output data will remain in the state before the control command was called. Only after the “Control commands” data exchange has been completed, will the I/O data be refreshed.

⚠ Caution!
Only those values specified for the command code should be used.

Check the values that you write in order to avoid malfunctions.

Requirement:
The “Control commands 9 byte” module must have been selected.

The master initiates the data exchange of the control commands and the addressed slave responds.

During communication 9 data bytes (byte 0 = toggle byte, bytes 1 to 8 information bytes) are sent via PROFIBUS.

The basic telegram structure is shown in the following diagram.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8

Byte 0 – Toggle byte

Byte 0 is used to activate the sending of a control command with the toggle function.

Bit	7	6	5	4	3	2	1	0
$01_{hex}/861_{hex}$	0/1	0	0	0	0	0	0	1
	Toggle bit	fixed						

Procedure

▶ To send a command, bit 7 must be toggled, i.e. set either from 1 to 0 or from 0 to 1.

▶ Then poll the toggle bit for the coupling modules response until it has the same status as the toggle bit sent. This status indicates to the master that the response to the sent command is valid.

▶ Do not send a new command until you have received a response (changing of the toggle bit), otherwise the response of the previous command will be overwritten before it can be read.

→

> In order to use input/output data and control commands simultaneously:
>
> Only after the “Control commands” data exchange has been completed, will the I/O data be refreshed.
>
> All specified commands and parameters must be transferred in hexadecimal format.

The following tables show the different control commands possible. These essential control commands fall into three essential categories – real-time clock, image and function blocks.

Read/write date and time

→ Please also note the relevant description of the real-time clock provided in the EZ800 manual and the EZD manual.

The latest edition of these manuals are available as PDF files from the Internet at: www.EatonElectrical.com. EZ800 manual search term: MN05013004E. EZD manual search term: MN05013005E.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command		
	Read	93	–
	Write	B3	–
	Response		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Len	05	05
3	Index	00	00
4 – 8	Data 1 – 5		
	Read operation	00	→ Table 43
	For write operation	→ Table 43	00

Table 43: Byte 4 – 8: Data 1 – 5

Byte	Contents		Value (hex)
4	Data 1	Hour (0 to 23)	00 – 17
5	Data 2	Minute (0 to 59)	00 – 3B
6	Data 3	Day (1 to 28; 29, 30, 31; depending on month and year)	01 – 1F
7	Data 4	Month (1 to 12)	01 – 0C
8	Data 5	Year (0 – 99, corresponds to 2000 – 2099)	00 – 63

Winter/summer time, DST

Telegram structure

Byte	Meaning		Value (hex), sent by	
			Master	Slave
0	Toggle byte		→ page 94	
1	Command			
		Read	93	–
		Write	B3	–
	Response			
		Read successful	–	C2
		Write successful	–	C1
		Command rejected	–	C0
2	Len		05	05
3	Index		01	01
4 – 8	Data 1 – 5			
		Read operation	00	→ Table 44
		For write operation	→ Table 44	00

Table 44: Byte 4 – 8: Data 1 – 5

Byte	Contents		Value (hex)
4	Data 1	Area	
		None	00
		Manual	01
		Automatic EU	02
		Automatic GB	03
		Automatic US	04
5	Data 2[1]	Set summer time day (1 to 28, 29, 30, 31 depending on month and year)	00 – 3B
6	Data 3[1]	Set Summer time month (1 to 12)	01 – 1F
7	Data 4[1]	Set winter time day (1 to 28, 29, 30, 31 depending on month and year)	01 – 0C
8	Data 5[1]	Set winter time month (1 to 12)	00 – 63

[1] The additional parameters Data 2 to Data 5 for automatic DST change are only relevant if you have set the “Manual” parameter for Data 1.

Example

The real-time clock of the EZ800 is to be set to Friday 23.05.2003, 14:36.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Write	B3	–
	Response: Write successful	–	C1
2	Len	05	05
3	Index	00	00
4	Data 1	0E	00
5	Data 2	24	00
6	Data 3	17	00
7	Data 4	05	00
8	Data 5	03	00

→ All values must be transferred as hexadecimal values.

Read/write image data

→ Please also observe the relevant description of possible image data provided in the EZ800 manual, the EZD manual, or in the EZSoft Help.

The latest editions of these manuals are available as PDF files from the Internet at: www.EatonElectrical.com. EZ800 manual search term: MN05013004E. EZD manual search term: MN05013005E.

The information provided in Section “General notes on working with image data” on Page 49 also applies to EZ800/EZD.

Overview

Operands	Meaning	Read/write	Command	Page
IW0	Read local inputs IW0	Read	01	100
IW1 – IW8	Read inputs of the stations IW1 to IW8	Read	01	102
IA1 – IA4	Read local analog inputs IA1 to IA4	Read	02	103
ID1 – ID16	Read local diagnostics ID1 to ID16	Read	03	105
QW0, QW1 – QW8	Read and write local QW0 outputs/outputs of the stations QW1 to QW8	Read/write	04	107
QA1	Reading and writing local analog output QA1	Read/write	05	109
P1 – P4	Reading local P buttons	Read	06	110
R1 – R16 S1 – S8	Reading RW.. inputs/SW.. outputs from EZ-LINK	Read	07/09	112
RN1 – RN32 SN1 – SN32	Reading receive data network RN1 .. RN32/send data network SN1 .. SN32	Read	08/0A	114
M...	Reading and writing markers	Read/write	0B – 0E	116

Read local inputs IW0

This command string enables you to read the local inputs of the EZ800/EZD. The relevant input word is stored in Intel format.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	01	01
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 45
6	Data 2 (High Byte)	00	→ Table 45
7 – 8	Data 3 – 4	00	00

Table 45: Byte 5 to 6: Data 1 to 2

Data 1	Bit 7	6	5	4	3	2	1	0
I1								0/1
I2							0/1	
..				..				
I8	0/1							
Data 2	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
I9								0/1
I10							0/1	
..				..				
I16	0/1							

Example: Read local inputs IW0

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Read	91	–
	Response: Read successful	–	C2
2	Len	02	02
3	Type	01	01
4	Index	00	00
5	Data 1	00	C4
6	Data 2	00	02
7	Data 3	00	00
8	Data 4	00	00

→ All values must be transferred as hexadecimal values.

The values Data 1 = C4 and Data 2 = 02 indicate that the inputs I8, I7, I3 and I10 have been set to 1.

Read inputs of the stations IW1 to IW8

The EZ800 and EZD devices can be remotely expanded very simply using the EZ-NET. The service offered here makes it possible to implement read access to the inputs of individual NET stations.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	01	01
4	Index	01 – 08[1]	01 – 08[1]
5	Data 1 (Low Byte)	00	→ Table 45 on Page 100.
6	Data 2 (High Byte)	00	
7 – 8	Data 3 – 4	00	00

[1] Corresponds to address of network station

Read local analog inputs IA1 to IA4

The analog inputs on the EZ800 and EZD basic units can be read directly via PROFIBUS-DP. The 16-bit value is transferred in Intel format (LowByte first).

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	02	02
4	Index	01 – 04[1]	01 – 04[1]
5	Data 1 (Low Byte)	00	See example
6	Data 2 (High Byte)	00	See example
7 – 8	Data 3 – 4	00	00

1) 01 = Analog input I7
02 = Analog input I8
03 = Analog input I11
04 = Analog input I12

Example

A voltage signal is present at analog input 1. The required telegrams for reading the analog value are as follows:

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Read	91	–
	Response: Read successful	–	C2
2	Len	02	02
3	Type	02	02
4	Index	01[1]	01[1]
5	Data 1	00	D9
6	Data 2	00	02
7	Data 3	00	00
8	Data 4	00	00

[1] 01 = Analog input 1

Byte 5 – Data 1 (Low Byte): $D9_{hex}$

Byte 6 – Data 2 (High Byte): 02_{hex}

→ corresponding 16-bit value: $02D9_{hex}$ = 729 (7.29 V)

Read local diagnostics ID1 to ID16

The local diagnostics (ID1 – ID8) bytes indicate the status of the individual NET stations. The connection to the remote station (only EZD) is indicated via ID9.

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	03	03
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 46
6	Data 2 (High Byte)	00	→ Table 46
7 – 8	Data 3 – 4	00	00

Table 46: Byte 5 to 6: Data 1 to 2

Data 1	Bit	7	6	5	4	3	2	1	0
ID1									0/1
ID2								0/1	
..					..				
ID8		0/1							
Data 2	Bit	7	6	5	4	3	2	1	0
ID9									0/1
–								1	
...					...				
–		1							

0/1= active/inactive NET station, –= not assigned

Example

Data 1 = F8, Data 2 = FF → In the EZ-NET network, the three stations are present with the NET IDs 1, 2, 3

Read and write local QW0 outputs/outputs of the stations QW1 to QW8

You can read and write the local outputs directly via PROFIBUS-DP. However, the outputs are only switched externally if the device is in Run mode and the addressed output is not being used in the circuit diagram. → Section “Read/write image data” on Page 99.

Telegram structure

Byte	Meaning		Value (hex), sent by	
			Master	Slave
0	Toggle byte		→ page 94	
1	Command			
		Read	91	–
		Write	B1	–
	Response			
		Read successful	–	C2
		Write successful	–	C1
		Command rejected	–	C0
2	Len		02	02
3	Type		04	04
4	Index[1]		00/01 – 08	00/01 – 08
5	Data 1			
		Read operation	00	→ Table 46
		For write operation	→ Table 47	00
6 – 8	Data 2 – 4		00	00

[1] 00 = Local output
01 – 08 = Outputs of network stations 1 – 8

Table 47: Byte5: Data

Data 1	Bit	7	6	5	4	3	2	1	0
Q1									0/1
Q2								0/1	
Q3							0/1		
Q4						0/1			
Q5					0				
Q6				0					
Q7			0						
Q8		0							

Reading and writing local analog output QA1

The commands provided can be used to access the local analog output of the EZ800 or EZD basic unit. When writing to the analog output, however, the value will only be output externally if the device concerned is in Run mode and the image concerned has not been overwritten by actual program. → Section “Read/write image data” on Page 99.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command		
	Read	91	–
	Write	B1	–
	Response		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Len	02	02
3	Type	05	05
4	Index	00	00
5 – 6	Data 1 – 2		
	Read operation	00	See example
	For write operation	See example	00
7 – 8	Data 3 – 4	00	00

Example

The analog output is to output a value of approx. 5 V.

500 = 01F4$_{hex}$	Byte 5 – Data 1 (LowByte) : F4$_{hex}$
	Byte 6 – Data 2 (HighByte): 01$_{hex}$

Reading local P buttons

The local P buttons are the display cursor buttons of the EZ800/EZD basic unit. You can scan the buttons in both Run and Stop mode.

→ Ensure that the P buttons are also activated via the SYSTEM menu (in the basic unit).

Only one byte has to be transferred for the P buttons.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	06	06
4	Index	00	00
5	Data 1 (Low Byte)	00	→ Table 48
6 – 8	Data 2 – 4	00	00

Table 48: Byte 5: Data

Data 1	Bit 7	6	5	4	3	2	1	0
P1								0/1
P2							0/1	
P3						0/1		
P4					0/1			
–				0				
–			0					
–		0						
–	0							

Reading RW.. inputs/SW.. outputs from EZ-LINK

This service allows you to read the local R and S data and the data of the NET stations (1 – 8) transferred via EZ-LINK, again from the relevant EZ800/EZD image.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	02	02
3	Type	For RW: 07	For RW: 07
		For SW: 09	For SW: 09
4	Index	00/01 – 08[1]	00/01 – 08[1]
5	Data 1 (Low Byte)	00	→ Table 49
6	Data 2 (High Byte)	00	→ Table 49
7 – 8	Data 3 – 4	00	00

1) 00 = Local input/output
01 – 08 = Address of network station (NET-ID 1 – 8)

Table 49: Byte 5 to 6: Data 1 to 2

Data 1		Bit 7	6	5	4	3	2	1	0
RW	SW								
R1	S1								0/1
R2	S2							0/1	
R3	S3						0/1		
R4	S4					0/1			
R5	S5				0/1				
R6	S6			0/1					
R7	S7		0/1						
R8	S8	0/1							
Data 2		Bit 7	6	5	4	3	2	1	0
R9	–								0/1
R10	–							0/1	
R11	–						0/1		
R12	–					0/1			
R13	–				0/1				
R14	–			0/1					
R15	–		0/1						
R16	–	0/1							

Reading receive data network RN1 .. RN32/send data network SN1 .. SN32

EZ-NET allows a point-to-point connection to be implemented between the individual NET stations. The RN and SN data are used for the data exchange (see the EZ800 manual and EZD manual).

The latest editions of these manuals are available as PDF files from the Internet at: www.EatonElectrical.com. EZ800 manual search term: MN05013004E. EZD manual search term: MN05013005E.

→ The RN SN data of the local device (Index = 0) to which the EZ204-DP is fitted cannot be scanned. In this case the command would be denied with the $0C_{hex}$ signal.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	91	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Len	04	04
3	Type	For RN1 – RN32: 08	
		For SN1 – SN32: 0A	
4	Index	01 – 08[1]	01 – 08[1]
5 – 8	Data 1 – 4	00	→ Table 50

[1] Corresponds to NET-ID

Table 50: Byte 5 to 8: Data 1 to 4

Data 1		Bit	7	6	5	4	3	2	1	0
RN1	SN1					...				0/1
...									0/1	
RN8	SN8		0/1							
Data 2		Bit	7	6	5	4	3	2	1	0
RN9	SN9									0/1
....						...				
RN16	SN16		0/1							
Data 3		Bit	7	6	5	4	3	2	1	0
RN17	SN17									0/1
...						...				
RN24	SN24		0/1							
Data 4		Bit	7	6	5	4	3	2	1	0
RN25	SN25									0/1
...						...				
RN32	SN32		0/1							

Reading and writing markers

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command		
	Read	91	–
	Write	B1	–
	Response		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Len	→ Table 51	→ Table 51
3	Type		
4	Index		
5 – 8	Data 1 – 4		
	Read operation	00	→ “Example 1” on Page 118
	For write operation	→ “Example 2” on Page 119	00

Table 51: Byte 2 – 4: Len, Type, Index

Operand				Len	Type	Index
Marker bit	M1	..	M96	01_{hex}	$0B_{hex}$	01 to 60_{hex}
Marker byte	MB1	..	MB96	01_{hex}	$0C_{hex}$	01 to 60_{hex}
Marker word	MW1	..	MW96	02_{hex}	$0D_{hex}$	01 to 60_{hex}
Marker double word	MD1	..	MD96	04_{hex}	$0E_{hex}$	01 to 60_{hex}

If required, refer to the more detailed description of the marker allocation in the EZ800 manual and EZD manual. Only a small extract of this manual is shown at this point in order to illustrate the allocation principle.

The latest editions of these manuals are available as PDF files from the Internet at: www.EatonElectrical.com. EZ800 manual search term: MN05013004E. EZD manual search term: MN05013005E.

Attention!
The function blocks and DW markers (32-bit values) of EZ800/EZD operate with signed values.

Applies to MD, MW, MB, M	Left = Most significant bit, byte, word			Right = Least significant bit, byte, word
32 bit	MD1			
16 bit	MW2		MW1	
8 bit	MB4	MB3	MB2	MB1
1 bit	M32 to M25	M24 to M17	M16 to M9	M8 to M1
32 bit	MD2			
16 bit	MW4		MW3	
8 bit	MB8	MB7	MB6	MB5
1 bit	M64 to M57	M56 to M49	M48 to M41	M40 to M33

→ The relevant marker values are transferred in Intel format. In other words, the first byte is the low byte (Byte 5) and the last byte the high byte.

Example 1

Read marker bit M62

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Read	91	–
	Response: Read successful	–	C2
2	Len	01	01
3	Type	0B	0B
4	Index	3E	3E
5	Data 1	00	01
6	Data 2	00	00
7	Data 3	00	00
8	Data 4	00	00

Result: Data 1 = 01_{hex} → M62 was set

Example 2

Write marker word MW32 with 823

$823_{dec} = 337_{hex}$ → Data 1 = 37_{hex}, Data 2 = 03_{hex}

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Write	B1	–
	Response: Write successful	–	C1
2	Len	02	02
3	Type	0D	0D
4	Index	20	20
5	Data 1	37	00
6	Data 2	03	00
7	Data 3	00	00
8	Data 4	00	00

Read/write function block data

Please also note the relevant description of the function blocks provided in the EZ800 manual and the EZD manual.

The latest edition of these manuals are available as PDF files from the Internet at: www.EatonElectrical.com. EZ800 manual search term: MN05013004E. EZD manual search term: MN05013005E.

General notes

Always note the following when working with function blocks:

- The relevant data is transferred in Intel format. In other words, the first byte is the low byte (Byte 5) and the last byte (byte 8) the high byte.
- The maximum data length is 4 bytes. All values must be transferred in hexadecimal format.
- All 32-bit values are treated as signed values. When transferring 32-bit values, ensure that the appropriate value range is suitable for long integers, i.e. signed.
  32-bit value: –2147483648 .. 0 .. +2147483647

Overview

Operands	Meaning	Read/write	Type	Page
A01 – A32	“Analog value comparators A01 .. A32”	Read/write	11	122
AR01 – AR32	“Arithmetic function blocks AR01 .. AR32”	Read/write	12	124
BC01 – BC32	“Block compare function blocks BC01 .. BC32”	Read/write	25	126
BT01 – BT32	“Block transfer function blocks BT01 .. BT32”	Read/write	26	128
BV01 – BV32	“Boolean sequence function blocks BV01 .. BV32”	Read/write	13	130
C01 – C32	“Counters C01 .. C32”	Read/write	14	132
CF01 – CF04	“Frequency counters CF01 .. CF04”	Read/write	15	134
CH01 – CH04	“High-speed counters CH01 .. CH04”	Read/write	16	136
CI01 – CI02	“Incremental encoder counters CI01 .. CI02”	Read/write	17	138
CP01 – CP32	“Comparators CP01 .. CP32”	Read/write	18	140
D01 – D32	“Text output function blocks D01 ..D32”	Read/write	19	142
DB01 – DB32	“Data function blocks DB01 .. DB32”	Read/write	1A	145
DC01 – DC32	“PID controllers DC01 .. DC32”	Read/write	27	147
FT01 – FT32	“Signal smoothing filters FT01 .. FT32”	Read/write	28	150
GT01 – GT32	“Receive network data function blocks GT01 .. GT32”	Read	1B	152
HW01 – HW32	“7-day time switches HW01 .. HW32”	Read	1C	154
HY01 – HY32	“Year time switches HY01 .. HY32”	Read	1D	157
LS01 – LS32	“Value scaling function blocks LS01 .. LS32”	Read/write	29	160
MR01 – MR32	“Master reset function blocks MR01 .. MR32”	Read	0F	162
NC01 – NC32	“Numerical converters NC01 .. NC32”	Read/write	2A	164
OT01 – OT04	“Operating hours counters OT01 .. OT04”	Read/write	1E	166
PT01 – PT32	“Send network data function blocks PT01 .. PT32”	Read	1F	168
PW01 – PW02	“Pulse width modulation function blocks PW01 .. PW02”	Read/write	2B	170
SC01	“Synchronize clock function block SC01”	Read	20	172
ST01	“Set cycle time function block ST01”	Read/write	2C	173
T01 – T32	“Timing relays T01 .. T32”	Read/write	21	175
VC01 – VC32	“Value limitation function blocks VC01 .. VC32”	Read/write	2D	178

Analog value comparators A01 .. A32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	11	11
3	Instance	01 – 20	01 – 20
4	Index	→ Table 52	→ Table 52
5 – 8	Data 1 – 4	00	depending on index, → Table 53, 54

Table 52: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 53		×	
01	Mode, → Table 54		×	
02	Comparison value 1	I1	×	c[1]
03	Gain factor for I1 (I1 = F1 χ value)	F1	×	c[1]
04	Comparison value 2	I2	×	c[1]
05	Gain factor for I2 (I2 = F2 χ value)	F2	×	c[1]
06	Offset for value I1	OS	×	c[1]
07	Switching hysteresis for value I2 (the value of HY is for both positive and negative hysteresis.)	HY	×	c[1]

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 7 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 53: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	CY[1]	Q1[2]

1) Status 1 if the value range is exceeded

2) Status 1 if the condition is fulfilled (e.g. I1 < I2 with LT mode)

Table 54: Index 1 - Mode

Data 1 (hex)		
00	LT	Less than (I1 < I2)
01	EQ	Equal to (I1 = IGT)
02	GT	Greater than (I1 > I2)

Arithmetic function blocks AR01 .. AR32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	12	12
3	Instance	01 – 20	01 – 20
4	Index	→ Table 55	→ Table 55
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 56, 57
	Write operation	depending on index, → Table 56, 57	00

Table 55: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 56		×	
01	Mode, → Table 57		×	
02	First operand	I1	×	c[1]
03	Second operand	I2	×	c[1]
04	Result	QV	×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 4 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 56: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1
FB output Data 3		–	–	–	–	–	ZE[1]	CY[1]

1) Status 1 if the value of the function block output QV (the calculation result) equals zero

2) Status 1 if the value range is exceeded

Table 57: Index 1 - Mode

Data 1 (hex)		
00	ADD	Add (I1 + I2 = QV)
01	SUB	Subtract (I1 – I2 = QV)
02	MUL	Multiply (I1 × I2 = QV)
03	DIV	Divide (I1 : I2 = QV)

Block compare function blocks BC01 .. BC32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	25	25
3	Instance	01 – 20	01 – 20
4	Index	→ Table 58	→ Table 58
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 59, 60
	Write operation	depending on index, → Table 59, 60	00

Table 58: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 59		×	
01	Mode, → Table 60		×	
02	Source range 1	I1	×	c[1]
03	Target range 2	I2	×	c[1]
04	Number of elements to compare: 8 (max. 192 bytes)	NO	×	c[1]

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 4 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 59: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	EN[1]
FB output Data 3		–	–	–	–	EQ[2]	E3[3]	E2[4]	E1[5]

1) Activates the function block on status 1.

2) Status 1 if the data ranges are equal; status 0 if not equal

Error outputs

3) Status 1 if the number of elements exceeds the source or target range.

4) Status 1 if the source and target range overlap.

5) Status 1 if the source or target range are outside of the available marker range (offset error)

Table 60: Index 1 - Mode

Mode	Data 1 (hex)	Operating mode
	02	Compare (internal EZ status signal for Block Compare mode)

Block transfer function blocks BT01 .. BT32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	26	26
3	Instance	01 – 20	01 – 20
4	Index	→ Table 61	→ Table 61
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 62, 63
	Write operation	depending on index, → Table 62, 63	00

Table 61: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 62		×	
01	Mode, → Table 63		×	
02	Source range 1	I1	×	c[1]
03	Target range 2	I2	×	c[1]
04	Number of elements to compare: max. 192 bytes	NO	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 and 3 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte .. Data 2 - High Byte).

Table 62: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	T[1]
FB output Data 3		–	–	–	–	–	E3[2]	E2[3]	E1[4]

1) Transfer of the source address specified at I1 to the target address specified at I2 on rising edge.

Error outputs

2) Status 1 if the number of elements exceeds the source or target range.
3) Status 1 if the source and target range overlap.
4) Status 1 if the source or target range are outside of the available marker range (offset error)

Table 63: Index 1 - Mode

Data 1 (hex)	Operating mode
00	INI: Initializes the target range with a byte value stored at the source address.
01	CPY: Copies a data block from a source to a target range. Data block size is specified at NO.

Boolean sequence function blocks BV01 .. BV32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	13	13
3	Instance	01 – 20	01 – 20
4	Index	→ Table 64	→ Table 64
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 65, 66
	Write operation	depending on index, → Table 65, 66	00

Table 64: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 65		×	
01	Mode, → Table 66		×	
02	First operand	I1	×	c[1]
03	Second operand	I2	×	c[1]
04	Operation result	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 4 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 65: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1
FB output Data 3		–	–	–	–	–	–	ZE[1]

[1] Status 1 if the value of the function block output QV (the operation result) equals zero

Table 66: Index 1 - Mode

Data 1 (hex)		
00	AND	And sequence
01	OR	Or sequence
02	XOR	Exclusive Or sequence
03	NET	Inverts the individual bits of the value at I1. The inverted value is represented as a signed decimal value.

Counters C01 .. C32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	14	14
3	Instance	01 – 20	01 – 20
4	Index	→ Table 67	→ Table 67
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 68
	Write operation	depending on index, → Table 68	00

Table 67: Operand overview

Index (hex)	Operand		Value	Read	Write
00	Bit IO		→ Table 68	×	
01	Mode/Parameter		–	–	–
02	Upper setpoint	SH	In integer range from –2147483648 to +2147483647	×	c[1]
03	Lower setpoint	SL		×	c[1]
04	Preset actual value	SV		×	c[1]
05	Actual value in Run mode	QV		×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 5 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 68: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	SE[1]	D[2]	C[3]	RE[4]
FB output Data 3		–	–	–	–	ZE[5]	CY[6]	FB[7]	OF[8]

1) Transfer preset actual value on rising edge
2) Count direction: 0 = up counting, 1 = down counting
3) Count coil, counts on every rising edge
4) Reset actual value to zero
5) Zero: Status 1 if the value of the function block output QV (the counter status) equals zero
6) Carry: Status 1 if the value range is exceeded
7) Fall below: Status 1 if the actual value ≦ lower setpoint
8) Overflow: Status 1 if the actual value ≧ upper setpoint

Frequency counters CF01 .. CF04

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	15	15
3	Instance	01 – 04	01 – 04
4	Index	→ Table 69	→ Table 69
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 70
	Write operation	depending on index, → Table 70	00

Table 69: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 70		×	
01	Mode/Parameter		–	–
02	Upper setpoint	SH	×	c[1]
03	Lower setpoint	SL	×	c[1]
04	Actual value in Run mode	QV	×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 4 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 70: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	EN[1]
FB output Data 3		–	–	–	–	–	ZE[2]	FB[3]	OF[4]

1) Counter enable

2) Zero: Status 1 if the value of the function block output QV (the counter status) equals zero

3) Fall below: Status 1 if the actual value ≦ lower setpoint

4) Overflow: Status 1 if the actual value ≧ upper setpoint

High-speed counters CH01 .. CH04

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	16	16
3	Instance	01 – 04	01 – 04
4	Index	→ Table 71	→ Table 71
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 72
	Write operation	depending on index, → Table 72	00

Table 71: Operand overview

Index (hex)	Operand		Value	Read	Write
00	Bit IO		→ Table 72	×	
01	Mode/Parameter		–	–	–
02	Upper setpoint	SH	In integer range from –2147483648 to +2147483647	×	c[1]
03	Lower setpoint	SL		×	c[1]
04	Preset actual value	SV		×	c[1]
05	Actual value in Run mode	QV		×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 5 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 72: Index 0: Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	EN[1]	SE[2]	D[3]	RE[4]
FB output Data 3		–	–	–	–	ZE[5]	CY[6]	FB[7]	OF[8]

1) Counter enable

2) Transfer preset actual value on rising edge

3) Count direction: 0 = up counting, 1 = down counting

4) Reset actual value to zero

5) Zero: Status 1 if the value of the function block output QV (the counter status) equals zero

6) Carry: Status 1 if the value range is exceeded

7) Fall below: Status 1 if the actual value ≦ lower setpoint

8) Overflow: Status 1 if the actual value ≧ lower setpoint

Incremental encoder counters CI01 .. CI02

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	17	17
3	Instance	01 – 02	01 – 02
4	Index	→ Table 73	→ Table 73
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 74
	Write operation	depending on index, → Table 74	00

Table 73: Operand overview

Index (hex)	Operand	Value	Read	Write
00	Bit IO	Table 74	×	
01	Mode/Parameter	–	–	–
02	Upper setpoint SH	In integer range from –2147483648 to +2147483647	×	c[1]
03	Lower setpoint SL		×	c[1]
04	Preset actual value SV		×	c[1]
05	Actual value in Run mode QV		×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 5 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 74: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	EN[1]	SE[2]	RE[3]
FB output Data 3		–	–	–	–	ZE[4]	CY[5]	FB[6]	OF[7]

1) Counter enable
2) Transfer preset actual value on rising edge
3) Reset actual value to zero
4) Zero: Status 1 if the value of the function block output QV (the counter status) equals zero
5) Carry: Status 1 if the value range is exceeded
6) Fall below: Status 1 if the actual value ≦ lower setpoint
7) Overflow: Status 1 if the actual value ≧ lower setpoint

Comparators CP01 .. CP32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	18	18
3	Instance	01 – 20	01 – 20
4	Index	→ Table 75	→ Table 75
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 76
	Write operation	depending on index, → Table 76	00

Table 75: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, Table 76		×	
01	Mode/Parameter		–	–
02	Comparison value	I1	×	c[1]
03	Comparison value	I2	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 and 3 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 76: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1
FB output Data 3		–	–	–	–	GT[1]	EQ[2]	LT[3]

[1] greater than: Status 1 if the value at I1 is greater than value at I2 (I1 > I2)

[2] equal: Status 1 if the value at I1 is equal to value at I2 (I1 = I2)

[3] less than: Status 1 if the value at I1 is less than value at I2 (I1 < I2)

Text output function blocks D01 ..D32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	19	19
3	Instance	01 – 20	01 – 20
4	Index	→ Table 77	→ Table 77
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 78
	Write operation	depending on index, → Table 78	00

Table 77: Operand overview

Index (hex)	Operand	Read	Write
00	Bit IO, → Table 78	×	
01	Mode/Parameter	–	–
02	Text line 1, column 1 - 4	×	
03	Text line 1, column 5 - 8	×	
04	Text line 1, column 9 - 12	×	
05	Text line 1, column 13 - 16	×	
06	Text line 2, column 1 - 4	×	
07	Text line 2, column 5 - 8	×	
08	Text line 2, column 9 - 12	×	
09	Text line 2, column 13 - 16	×	
10	Text line 3, column 1 - 4	×	
11	Text line 3, column 5 - 8	×	
12	Text line 3, column 9 - 12	×	
13	Text line 3, column 13 - 16	×	
14	Text line 4, column 1 - 4	×	
15	Text line 4, column 5 - 8	×	
16	Text line 4, column 9 - 12	×	
17	Text line 4, column 13 - 16	×	
18	Variable 1	×	c[1]
19	Variable 2	×	c[1]
20	Variable 3	×	c[1]
21	Variable 4	×	c[1]
22	Scaling minimum value 1	×	
23	Scaling minimum value 2	×	
24	Scaling minimum value 3	×	
25	Scaling minimum value 4	×	
26	Scaling maximum value 1	×	

Index (hex)	Operand	Read	Write
27	Scaling maximum value 2	×	
28	Scaling maximum value 3	×	
29	Scaling maximum value 4	×	
30	Control information line 1	×	
31	Control information line 2	×	
32	Control information line 3	×	
33	Control information line 4	×	

1) The value can only be written if it is assigned to a constant in the program.

→ The variables 1 to 4 (index 18 to 21) are transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 78: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	EN[1]
FB output Data 3		–	–	–	–	–	–	–	Q1[2]

1) Text function block enable

2) Status 1, text function block is active

Data function blocks DB01 .. DB32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	1A	1A
3	Instance	01 – 20	01 – 20
4	Index	→ Table 79	→ Table 79
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 80
	Write operation	depending on index, → Table 80	00

Table 79: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, Table 80		×	
01	Mode/Parameter		–	–
02	Input value: value that is transferred to the QV output when the FB is triggered.	I1	×	c[1]
03	Output value	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 and 3 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 80: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	T[1]
FB output Data 3		–	–	–	–	–	–	–	Q1[2]

[1] Transfer of the value present at I1 on rising edge.

[2] Status 1 if the trigger signal is 1.

PID controllers DC01 .. DC32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	27	27
3	Instance	01 – 20	01 – 20
4	Index	→ Table 81	→ Table 81
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 82, 83
	Write operation	depending on index, → Table 82, 83	

Table 81: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 82		×	
01	Mode, → Table 83		×	
02	Setpoint: –32768 to +32767	I1	×	c[1]
03	Actual value: –32768 to +32767	I2	×	c[1]
04	Proportional gain [%], Value range: 0 to 65535	KP	×	c[1]
05	Reset time [0.1 s], Value range: 0 to 65535	TN	×	c[1]
06	Rate time [0.1 s], Value range: 0 to 65535	TV	×	c[1]
07	Scan time = Time between function block calls. Value range: 0.1s to 6553.5s. If 0 is entered as the value, the scan time will be determined by the program cycle time.	TC	×	c[1]
08	Manual manipulated variable, value range: –4096 to +4095	MV	×	c[1]
09	Manipulated variable • Mode: UNI, value range: 0 to +4095 (12 bit) • Mode: BIP, value range: –4096 to +4095 (13 bit)	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 and 9 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte .. Data 2 - High Byte).

Table 82: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	SE[1]	ED[2]	EI[3]	EP[4]	EN[5]
FB output Data 3		–	–	–	–	–	–	–	LI[6]

1) Transfer of manual manipulated variable on status 1
2) Activation of D component on status 1
3) Activation of I component on status 1
4) Activation of P component on status 1
5) Activates the function block on status 1.
6) Status 1 if the value range of the medium-voltage was exceeded

Table 83: Index 1 - Mode

Data 1	Operating mode
UNP unipolar	The manipulated variable is output as a unipolar 12-bit value. Corresponding value range for QV 0 to 4095.
BIP bipolar	The manipulated variable is output as a bipolar 13-bit value. Corresponding value range for QV –4096 to 4095

Signal smoothing filters FT01 .. FT32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	28	28
3	Instance	01 – 20	01 – 20
4	Index	→ Table 84	→ Table 84
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 85
	Write operation	depending on index, → Table 85	00

Table 84: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 85		×	
01	Mode/Parameter		–	–
02	Input value, value range: –32768 to +32767	I1	×	c[1]
03	Recovery time [0.1 s], Value range: 0 to 65535	TG	×	c[1]
04	Proportional gain [%], Value range: 0 to 65535	KP	×	c[1]
05	Delayed output value, value range: –32768 to +32767	QV	×	

1) The value can only be written if it is assigned to a constant in the program.

Table 85: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	EN[1]

1) Activates the function block on status 1.

Receive network data function blocks GT01 .. GT32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	1B	1B
3	Instance	01 – 20	01 – 20
4	Index	→ Table 86	
5 – 8	Data 1 – 4	00	depending on index, → Table 87, 88

Table 86: Operand overview

Index (hex)	Operand	Read	Write
00	Bit IO,→ Table 87	×	
01	Mode/Parameters, → Table 88	×	–
02	Output value: actual value from the network QV	×	

→ The data for index 2 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 87: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	Q[1]

1) Status 1 if a new value is present that is transferred from the NET network.

Table 88: Index 1 – Mode/Parameters (designation of PUT FB with data to be received)

Mode	Data 1	NET-ID[1]	
		0	NET-ID 1
		..	..
		7	NET-ID 8
Parameters	Data 3	Instance[2]	
		0	PT01
		..	..
		31	PT32

1) Number of station sending the value. Possible station numbers: 01 to 08

2) Send FB (e.g. PT 20) of the sending NET station. Possible station numbers: 01 – 32

7-day time switches HW01 .. HW32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	1C	1C
3	Instance	01 – 20	01 – 20
4	Index	→ Table 89	
5 – 8	Data 1 – 4	00	depending on index, → Table 90

Table 89: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO →	Table 90	×	
01	Mode/Parameter		–	–
02	Parameters →	Table 91	×	
	Channel A			
03	Channel B			
04	Channel C			
05	Channel D			

Table 90: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	Q[1]

[1] Status 1 if the switch-on condition is fulfilled.

The data in the following table is shown in the Motorola format although it is actually transferred in Intel format.

Table 91: Index 2 – 5, Parameter channels A – D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								Date 1							
ON	d4	d3	d2	d1	d0	h4	h3	h2	h1	h0	m5	m4	m3	m2	m1	m0
	Weekday					Hour					Minute					

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 4								Date 3							
OFF	d4	d3	d2	d1	d0	h4	h3	h2	h1	h0	m5	m4	m3	m2	m1	m0
	Weekday					Hour					Minute					

m5 to m0: Minute (0 to 59)
h4 to h0: Hour (0 to 23)
d5 to d0: Weekday (0 = Sunday to 6 = Saturday)

Example

The channel A parameters of 7-day time switch HW19 are to be read.

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Read	92	–
	Response: Read successful	–	C2
2	Type	1C	1C
3	Instance	13	13
4	Index	02	02
5	Data 1	00	62
6	Data 2	00	0B
7	Data 3	00	7B
8	Data 4	00	25

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2 = $0B_{\text{hex}}$								Date 1 = 62_{hex}							
ON	0	0	0	0	1	0	1	1	0	1	1	0	0	0	1	0
	Weekday					Hour					Minute					

Switch-on time:
Weekday = 01_{hex} .. Monday
Hour = $0D_{\text{hex}}$ .. 1300 hours
Minute = 22_{hex} .. 34 minutes

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 4 = 25_{hex}								Date 3 = $7B_{\text{hex}}$							
OFF	0	0	1	0	0	1	0	1	0	1	1	1	1	0	1	1
	Weekday					Hour					Minute					

Switch-off time:
Weekday = 04_{hex} .. Thursday
Hour = 15_{hex} .. 2100 hours
Minute = 59_{hex} .. 34 minutes

Year time switches HY01 .. HY32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	1D	1D
3	Instance	01 – 20	01 – 20
4	Index	→ Table 92	
5 – 8	Data 1 – 4	00	depending on index, → Table 93

Table 92: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO	→ Table 93	×	
01	Mode/Parameter		–	–
02	Parameters Channel A	→ Table 94	×	
03	Channel B			
04	Channel C			
05	Channel D			

Table 93: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	Q[1]

[1] Status 1 if the switch-on condition is fulfilled.

The data in the following table is shown in the Motorola format although it is actually transferred in Intel format.

Table 94: Index 2 – 5, Parameter channels A – D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								Date 1							
ON	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Year							Month				Day				

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 4								Date 3							
OFF	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Year							Month				Day				

d4 ... d0: Day (1 .. 31), m3 ... m0: Month (1 .. 12), y6 ... y0: Year (0: 2000 .. 99: 2099)

Example

The channel A parameters of year time switch HY14 are to be written.

Index 2 – 5, Parameter channels A – D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								Date 1							
ON	0	0	0	0	0	1	1	0	1	1	0	0	1	1	1	0
	Year							Month				Day				

Switch-on time:
Day = 14 = $0E_{hex}$ = 0000 1110b
Month = 6 (June) = 06_{hex} = 0000 0110b
Year = 2003 = 03_{hex} = 0000 0011b

Index 2 – 5, Parameter channels A – D

Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 2								Date 1							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
	Date 4								Date 3							
OFF	y6	y5	y4	y3	y2	y1	y0	m3	m2	m1	m0	d4	d3	d2	d1	d0
	Year							Month				Day				

Switch-off time:
Day = 3 = 03_{hex} = 0000 0011b
Month = 10 (October) = $0A_{hex}$ = 0000 1010b
Year = 2012 = $0C_{hex}$ = 0000 1100b

Resulting telegram:

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	81	80
1	Command: Write	B2	–
	Response: Write successful	–	C1
2	Type	1D	1D
3	Instance	0E	0E
4	Index	02	02
5	Data 1	8E	00
6	Data 2	06	00
7	Data 3	43	00
8	Data 4	19	00

Value scaling function blocks LS01 .. LS32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	29	29
3	Instance	01 – 20	01 – 20
4	Index	→ Table 95	→ Table 95
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 96
	Write operation	depending on index, → Table 96	

Table 95: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 96		×	
01	Mode/Parameter		–	–
02	Input value, value range: 32 bit	I1	×	c[1]
03	Interpolation point 1, X coordinate, value range: 32 bit	X1	×	c[1]
04	Interpolation point 1, Y coordinate, value range: 32 bit	Y1	×	c[1]
05	Interpolation point 2, X coordinate, value range: 32 bit	X2	×	c[1]
06	Interpolation point 2, Y coordinate, value range: 32 bit	Y2	×	c[1]
07	Output value: contains the scaled input value	QV	×	

1) The value can only be written if it is assigned to a constant in the program.

Table 96: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	EN[1]

1) Activates the function block on status 1.

Master reset function blocks MR01 .. MR32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	0F	0F
3	Instance	01 – 20	01 – 20
4	Index		
	Bit IO	00	00
	Mode	01	01
5 – 8	Data 1 – 4	00	depending on index, → Table 97, 98

Table 97: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	T[1]
FB output Data 3		–	–	–	–	–	–	–	Q1[2]

1) Trigger coil. The appropriate Reset is executed if the coil is triggered (with a rising edge).

2) Status 1 if the trigger coil MR..T is 1.

Table 98: Index 1 - Mode

Data 1 (hex)		
00	Q	Outputs Q.., *Q.., S.., *S.., *SN.., QA01 are reset to 0. * depending on the NET-ID
01	M	The marker range MD01 to MD48 is reset to 0.
02	ALL	Has an effect on Q and M.

Numerical converters NC01 .. NC32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	2A	2A
3	Instance	01 – 20	01 – 20
4	Index	→ Table 99	→ Table 99
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 100, 101
	Write operation	depending on index, → Table 100, 101	00

Table 99: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, Table 100		×	
01	Mode, Table 101		×	
02	Input value: operand to be converted	I1	×	c[1]
03	Output value: contains the conversion result	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 and 3 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte .. Data 2 - High Byte).

Table 100: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	EN[1]

[1] Activates the function block on status 1.

Table 101: Index 1 - Mode

Data 1 (hex)		
00	BCD	Converts a BCD coded decimal value to an integer value.
01	BIN	Converts an integer value to a BCD coded decimal value.

Operating hours counters OT01 .. OT04

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	1E	1E
3	Instance	01 – 04	01 – 04
4	Index	→ Table 102	→ Table 102
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 103
	Write operation	depending on index, → Table 103	00

Table 102: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 103		×	
01	Mode/Parameter		–	–
02	Upper threshold value	I1	×	c[1]
03	Actual value of operating hours counter	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

Table 103: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	RE[1]	EN[2]
FB output Data 3		–	–	–	–	–	–	–	Q1[3]

[1] Reset coil: Status 1 resets the counter actual value to zero.

[2] Enable coil

[3] Status 1 if the setpoint was reached (greater than/equal to)

→ The data for index 2 and 3 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Send network data function blocks PT01 .. PT32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	1F	1F
3	Instance	01 – 20	01 – 20
4	Index	→ Table 104	
5 – 8	Data 1 – 4	00	depending on index, → Table 105

Table 104: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 105		×	
01	Mode/Parameter		–	–
02	Input value: Setpoint that it transmitted to the NET network	I1	×	

→ The data for index 2 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 105: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	T[1]
FB output Data 3		–	–	–	–	–	–	–	Q1[2]

1) Trigger coil. The value is provided on the NET if the coil is triggered (with a rising edge).

2) Status 1 if the trigger coil PT..T_ is also 1.

Pulse width modulation function blocks PW01 .. PW02

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	2B	2B
3	Instance	01 – 02	01 – 02
4	Index	→ Table 106	→ Table 106
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 107
	Write operation	depending on index, → Table 107	00

Table 106: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 107		×	
01	Mode/Parameter		–	–
02	Manipulated variable, value range: 0 to 4095 (12 bit)	SV	×	c[1]
03	Period duration [ms], Value range: 0 to 65535	PD	×	c[1]
04	Minimum on duration [ms], Value range: 0 to 65535	ME	×	c[1]

1) The value can only be written if it is assigned to a constant in the program.

Table 107: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	EN[1]
FB output Data 3		–	–	–	–	–	–	–	E1[2]

1) Activates the function block on status 1.

2) Status 1 if below the minimum on duration or minimum off duration

Synchronize clock function block SC01

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	
1	Command: Read	92	–
	Response:		
	Read successful	–	C2
	Command rejected	–	C0
2	Type	20	20
3	Instance	01	01
4	Index	→ Table 108	
5 – 8	Data 1 – 4	00	depending on index, → Table 109

Table 108: Operand overview

Index (hex)	Operand	Read	Write
00	Bit IO, → Table 109	×	
01	Mode/Parameter	–	–

Table 109: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	–	–	T[1]
FB output Data 3		–	–	–	–	–	–	–	Q1[2]

1) Trigger coil. If the coil is triggered (rising edge), the current date, weekday and time of the sending station are automatically sent to the NET network.

2) Status 1 if the trigger coil SC01T_ is also 1.

Set cycle time function block ST01

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	2C	2C
3	Instance	01	01
4	Index	→ Table 110	→ Table 110
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 111
	Write operation	depending on index, → Table 111	00

Table 110: Operand overview

Index (hex)	Operand	Read	Write
00	Bit IO, → Table 111	×	
01	Mode/Parameter	–	–
02	Cycle time in ms, I1 value range: 0 – 1000	×	c[1]

[1] The value can only be written if it is assigned to a constant in the program.

Table 111: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	EN[1]

[1] Activates the function block on status 1.

Timing relays T01 .. T32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	21	21
3	Instance	01 – 20	01 – 20
4	Index	→ Table 112	→ Table 112
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 113, 114
	Write operation	depending on index, → Table 113, 114	

Table 112: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 113		×	
01	Mode/Parameters, → Table 114		×	
02	Setpoint 1: Time setpoint 1	I1	×	c[1]
03	Setpoint 2: Time setpoint 2 (with timing relay with 2 setpoints)	I2	×	c[1]
04	Actual value: Time elapsed in Run mode	QV	×	

1) The value can only be written if it is assigned to a constant in the program.

→ The data for index 2 to 4 is transferred as a 32-bit value in Intel format (Data 1 – Low Byte to Data 4 – High Byte).

Table 113: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB input Data 1		–	–	–	–	–	ST[1]	EN[2]	RE[3]
FB output Data 3		–	–	–	–	–	–	–	Q1[4]

1) Stop, the timing relay is stopped (Stop coil)

2) Enable, the timing relay is started (trigger coil)

3) Reset, the timing relay is reset (reset coil)

4) Switch contact

Table 114: Index 1 - Mode/Parameters

Mode	Data 1	Operating mode
	0	On-delayed
	1	On-delayed with random setpoint
	2	Off-delayed
	3	Off-delayed with random setpoint
	4	On and off delayed (two time setpoints)
	5	On and off delayed each with random setpoint (two time setpoints)
	6	Impulse transmitter
	7	Flashing relay (two time setpoints)
	8	Off-delayed, retriggerable
	9	Off-delayed with random setpoint, retriggerable
Parameters	Data 3	Operating mode
	0	S (milliseconds)
	1	M:S (seconds)
	2	H:M (minutes)

Value limitation function blocks VC01 .. VC32

Telegram structure

Byte	Meaning	Value (hex), sent by	
		Master	Slave
0	Toggle byte	→ page 94	→ page 94
1	Command:		
	Read	92	–
	Write	B2	–
	Response:		
	Read successful	–	C2
	Write successful	–	C1
	Command rejected	–	C0
2	Type	2D	2D
3	Instance	01 – 20	01 – 20
4	Index	→ Table 115	→ Table 115
5 – 8	Data 1 – 4		
	Read operation	00	depending on index, → Table 116
	Write operation	depending on index, → Table 116	

Table 115: Operand overview

Index (hex)	Operand		Read	Write
00	Bit IO, → Table 116		×	
01	Mode/Parameter		–	–
02	Input value	I1	×	c[1]
03	Upper limit value	SH	×	c[1]
04	Lower limit value	SL	×	c[1]
05	Output value: outputs the value present at input I1 within the set limits.	QV	×	

[1] The value can only be written if it is assigned to a constant in the program.

Table 116: Index 0 – Bit IO

	Bit	7	6	5	4	3	2	1	0
FB output Data 3		–	–	–	–	–	–	–	EN[1]

[1] Activates the function block on status 1.

Appendix

What Happens If ...?

Event	Explanation	Remedy
POW LED not lit	No power supply	Connect and switch on power supply
POW LED flashing	Data transfer via EZ-LINK OK	
BUS LED not lit	No PROFIBUS-DP data communication	Connect and start PROFIBUS-DP
BUS LED lit	Data transfer via PROFIBUS-DP OK	
Slave not signalling	– No slave address set – No bus terminating resistor present – Cable, plug faulty – No power supply	– Set slave address – Set bus terminating resistors – Check connection – Provide power supply to device
Write command rejected	– Command not permissible – EZ display not on the Status display	– Change command – Show Status display
Actual value is zero	No actual value present	Function relay does not have an actual value or not triggered

Overview of commands

EZ700

The commands are sorted in ascending order:

Command value hex	
01	Write T1 timing relay setpoint
02	Write T2 timing relay setpoint
03	Write T3 timing relay setpoint
04	Write T4 timing relay setpoint
05	Write T5 timing relay setpoint
06	Write T6 timing relay setpoint
07	Write T7 timing relay setpoint
08	Write T8 timing relay setpoint
09	Write C1 counter relay setpoint
0A	Write C2 counter relay setpoint
0B	Write C3 counter relay setpoint
0C	Write C4 counter relay setpoint
0D	Write C5 counter relay setpoint
0E	Write C6 counter relay setpoint
0F	Write C7 counter relay setpoint
10	Write C8 counter relay setpoint
12	Write time switch 1 channel A
13	Write time switch 1 channel B
14	Write time switch 1 channel C
15	Write time switch 1 channel D
16	Write time switch 2 channel A
17	Write time switch 2 channel B
18	Write time switch 2 channel C
19	Write time switch 2 channel D
1A	Write time switch 3 channel A

Command value hex	
1B	Write time switch 3 channel B
1C	Write time switch 3 channel C
1D	Write time switch 3 channel D
1E	Write time switch 4 channel A
1F	Write time switch 4 channel B
20	Write time switch 4 channel C
21	Write time switch 4 channel D
22	Write analog value comparator A1
23	Write analog value comparator A2
24	Write analog value comparator A3
25	Write analog value comparator A4
26	Write analog value comparator A5
27	Write analog value comparator A6
28	Write analog value comparator A7
29	Write analog value comparator A8
2A	Write time
2B	Read time switch 1 channel A
2C	Read time switch 1 channel B
2D	Read time switch 1 channel C
2E	Read time switch 1 channel D
2F	Read time switch 2 channel A
30	Read time switch 2 channel B
31	Read time switch 2 channel C
32	Read time switch 2 channel D
33	Read time switch 3 channel A
34	Read time switch 3 channel B
35	Read time switch 3 channel C
36	Read time switch 3 channel D

Command value hex	
37	Read time switch 4 channel A
38	Read time switch 4 channel B
39	Read time switch 4 channel C
3A	Read time switch 4 channel D
3C	Read time
3D	Read status of analog and digital inputs
3E	Read status of P buttons and operator buttons
3F	Read status of timing relays, counter relays, time switches and analog value comparators
40	Read status of markers, digital outputs and text display markers
41	Read T1 actual value
42	Read T2 actual value
43	Read T3 actual value
44	Read T4 actual value
45	Read T5 actual value
46	Read T6 actual value
47	Read T7 actual value
48	Read T8 actual value
49	Read C1 counter relay actual value
4A	Read C2 counter relay actual value
4B	Read C3 counter relay actual value
4C	Read C4 counter relay actual value
4D	Read C5 counter relay actual value
4E	Read C6 counter relay actual value
4F	Read C7 counter relay actual value
50	Read C8 counter relay actual value

EZ800/EZD

Date and time	Byte 1 Command (hex)	Byte 2 Len1 (hex)	Byte 3 Index (hex)
Read/write date and time	93/B3	05	00
Winter/summer time, DST			01

Image data	Byte 1 Command (hex)	Byte 2 Len1 (hex)	Byte 3 Type (hex)	Byte 4 Index (dec)
Read/write image data	91/B1			
Local inputs: I1 – I16		2	01	0
Read inputs of the stations IW1 to IW8				1 – 8
Read local analog inputs IA1 to IA4			02	1 – 4
Read local diagnostics ID1 to ID16			03	0
Read and write local QW0 outputs/outputs of the stations QW1 to QW8			04	0/1 – 8
Reading and writing local analog output QA1			05	0
Reading local P buttons		1	06	0
Reading RW.. inputs/SW.. outputs from EZ-LINK		2	07/09	0
Reading receive data network RN1 .. RN32/send data network SN1 .. SN32				1 – 8
Reading receive data network RN1 .. RN32/send data network SN1 .. SN32		4	08/0A	1 – 8
Marker bit M1 .. M96		1	0B	1 – 96
Marker byte MB1 .. MB96			0C	1 – 96
Marker word MW1 .. MW96		2	0D	1 – 96
Marker double word MD1 .. MD96		4	0E	1 – 96

1) Len... stands for the number of data bytes to be sent.

Function blocks	Byte 1 Command (hex)	Byte 2 Type (hex)	Byte 3 Instance (hex)
Read/write function blocks	92/B2		
Receive network data function blocks GT01 .. GT32		0F	1 – 20
Analog value comparators A01 .. A32		11	1 – 20
Arithmetic function blocks AR01 .. AR32		12	1 – 20
Boolean sequence function blocks BV01 .. BV32		13	1 – 20
Counters C01 .. C32		14	1 – 20
Frequency counters CF01 .. CF04		15	1 – 20
High-speed counters CH01 .. CH04		16	1 – 4
Incremental encoder counters CI01 .. CI02		17	1 – 2
Comparators CP01 .. CP32		18	1 – 20
Text output function blocks D01 ..D32		19	1 – 20
Data function blocks DB01 .. DB32		1A	1 – 20
Receive network data function blocks GT01 .. GT32		1B	1 – 20
7-day time switches HW01 .. HW32		1C	1 – 20
Year time switches HY01 .. HY32		1D	1 – 20
Operating hours counters OT01 .. OT04		1E	1 – 4
Send network data function blocks PT01 .. PT32		1F	1 – 20
Synchronize clock function block SC01		20	1
Set cycle time function block ST01		21	1 – 20
Block compare function blocks BC01 .. BC32		25	1 – 20
Block transfer function blocks BT01 .. BT32		26	1 – 20
PID controllers DC01 .. DC32		27	1 – 20
Signal smoothing filters FT01 .. FT32		28	1 – 20
Value scaling function blocks LS01 .. LS32		29	1 – 20
Numerical converters NC01 .. NC32		2A	1 – 20
Pulse width modulation function blocks PW01 .. PW02		2B	1 – 2
Set cycle time function block ST01		2C	1
Value limitation function blocks VC01 .. VC32		2D	1 – 20

Technical Data

General

Standards and regulations	EN 55011, EN 55022, IEC/EN 61-4, IEC 60068-2-27, IEC 61158
Dimensions (W × H × D)	35.5 × 90 × 56.5 mm
Weight	150 g
Mounting	Top-hat rail to DIN 50022, 35 mm Screw fixing with fixing brackets EZB4-101-GF1 (accessories)

Ambient temperatures

Ambient temperature Installed horizontally/vertically	Cold to IEC 60068-2-1 Heat to IEC 60068-2-2	–25 to 55 °C
Condensation		Prevent condensation with suitable measures
Storage/transport temperature		–40 to +70 °C
Relative air humidity	IEC 60068-2-30	5 to 95 %, non-condensing
Air pressure (operation)		795 to 1080 hPa
Corrosion resistance	IEC 60068-2-42 IEC 60068-2-43	SO_2 10 cm^3/m^3, 4 days H_2S 1 cm^3/m^3, 4 days

Ambient mechanical conditions

Pollution degree		2
Degree of protection	EN 50178 IEC 60529 VBG4	IP20
Oscillations	IEC 60068-2-6	10 to 57 Hz (constant amplitude 0.15 mm) 57 to 150 Hz (constant acceleration 2 g)
Shocks	IEC 60068-2-27	18 shocks (semi-sinusoidal 15 g/11 ms)
Drop	IEC 60068-2-31	Drop height 50 mm
Free fall, packaged	IEC 60068-2-32	1 m

Electromagnetic compatibility (EMC)

Electrostatic discharge	IEC/EN 61000-4-2, degree of severity 3	8 kV air discharge, 6 kV contact discharge
Electromagnetic fields	IEC/EN 61000-4-3	Field strength 10 V/m
Radio interference suppression	EN 55011, EN 55022	Limit class A
Burst	IEC/EN 61000-4-4, degree of severity 3	2 kV supply lines, 1 kV signal lines
High-energy pulses (surge)		
EZ...-DC...	IEC/EN 61000-4-5, degree of severity 2	0.5 kV power cable symmetrical
Line-conducted interference	IEC/EN 61000-4-6	10 V

Dielectric strength

Measurement of the clearance and creepage distance	EN 50178, UL 508, CSC C22.2 No 142
Dielectric strength	EN 50178

Tools and cable cross-sections

Solid	
min.	0.2 mm^2 , AWG 22
max.	4 mm^2, AWG 12
Flexible with ferrule	
min.	0.2 mm^2 , AWG 22
max.	2.5 mm^2, AWG 12
Slot-head screwdriver, width	3.5 × 0.8 mm
Tightening torque max.	0.5 Nm

Power supply

Rated voltage	
Rated value	24 V DC, –15 %, +20 %
Permissible range	20.4 to 28.8 V DC
Residual ripple	< 5 %
Input current at 24 V DC	Normally 200 mA
Voltage dips (IEC/EN 61131-2)	10 ms
Power dissipation at 24 V DC	Normally 4.8 W

LEDs

Power LED (POW)	green
PROFIBUS-DP LED (BUS)	green

	PROFIBUS-DP
Device connection	SUB-D 9-pole, socket
Electrical isolation	Bus to power supply (simple) Bus and power supply to EZ basic unit (safe isolation)
Function	PROFIBUS-DP slave
Interface	RS 485
Bus protocol	PROFIBUS-DP
Baud rates	Automatic search up to 12 MBd
Bus terminating resistors	Connectable via plug
Bus addresses	1 to 126 addressable via EZ basic unit with display or EZSoft
Services	
Inputs module	All data S1 to S8 (EZ/EZD)
Outputs module	All data R1 to R16 (EZ/EZD)
Control commands module	Read/Write Time, day, summer/winter time (DST) All parameters of the EZ function relays

Dimensions

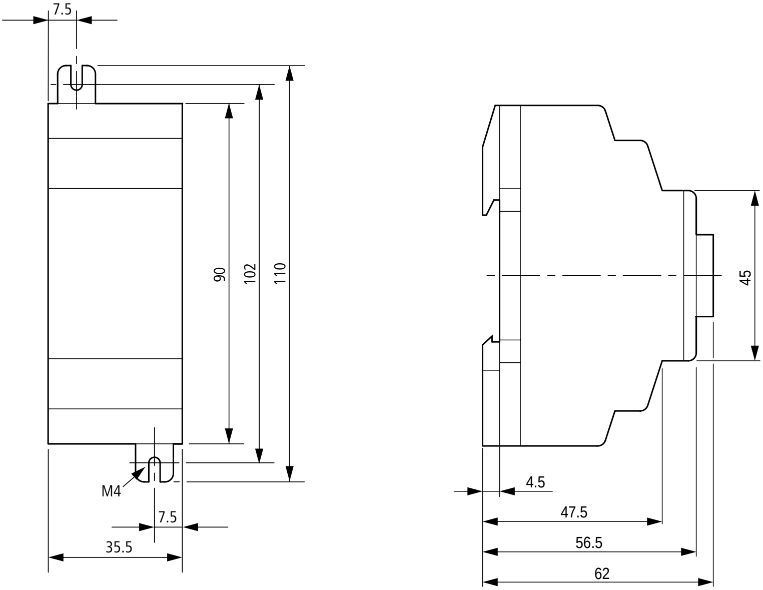


Figure 9: Dimensions EZ204-DP (mm)

Glossary

This glossary refers to subjects relating to PROFIBUS-DP.

Acknowledge
Acknowledgement returned by the receiving station after having received a signal.

Active metal component
Conductor or conductive component that is live when in operation.

Address
Number, for example, for identifying a memory location, a system or a module within a network.

Addressing
Assignment or setting of an address such as for a module in a network.

Analog
Value, such as voltage, that is infinitely variable and proportional. Analog signals can acquire any value within specific limits.

Automation device
Control device with inputs and outputs that is connected to a technical process. Programmable controllers (PLCs) are a special group of automation devices.

Baud
Unit for the data transfer rate. One baud corresponds to the transmission of one bit per second (bit/s).

Baud rate
Unit of measure of the data transmission speed in bit/s.

Bidirectional
Operation in both directions.

Bus
Bus cable system for data exchange between CPU, memory and I/O level. A bus can consist of several parallel segments, such as the data bus, address bus, control bus and power supply bus.

Bus cycle time
Time interval in which a master will serve all slaves or stations in a bus system, i.e. writes their outputs and reads their inputs.

Bus line
Smallest unit connected to the bus. Consists of the PLC, a module and a bus interface for the module.

Bus system
The entirety of all units which communicate across a bus.

Bus terminating resistor

Resistor at the beginning and end of a bus line for preventing disturbance caused by signal reflections and for adapting bus cables. Bus terminating resistors must always be the last unit at the end of a bus segment.

Capacitive coupling

Capacitive (electrical) coupling develops between two conductors carrying different potentials. Typical interference sources are, for example parallel signal cables, contactor relays and static discharge.

Chassis ground

Entirety of all interconnected inactive equipment parts that do not have any contact voltage, even in the event of a fault.

Coding element

Two-part element for the unambiguous allocation of electronic and basic module.

Command-capable modules

Command-capable modules are modules with an internal memory that are capable of executing particular commands (such as output substitute values).

Common potential

Electrical interconnection of the reference potentials of the control and load circuit of I/O modules.

Configuring

Systematic arrangement of the I/O modules of a station.

CPU

Abbreviation for “Central Processing Unit”. Central unit for data processing, which represents the core element of a computer.

Digital

A value, for example voltage, that can only be represented by a certain number of states within a defined range, usually defined as 0 and 1.

DIN

Abbreviation for “Deutsches Institut für Normungen e.V.” (German Institute for Standardization).

Earthing strip

Flexible conductor, mostly braided. Interconnects inactive parts of equipment, e.g. the doors of a control panel and the switch cabinet body.

Electrical equipment

All objects that are used for the generation, conversion, transfer, distribution and use of electric power, such as conductors, cables, machines, control devices.

EMC

Abbreviation for “Electromagnetic Compatibility”. The ability of electrical equipment to function trouble-free within a particular environment without a negative effect on the environment concerned.

EN

Abbreviation for “European Norm” or European standard.

Equipotential bonding

Adaptation of the electrical level of the body of electrical equipment and auxiliary conductive bodies by means of an electrical connection.

ESD

Abbreviation for “Electrostatic Discharge”.

Field supply

Voltage supply to field devices as well as signal voltage.

Fieldbus

Data network on the sensor/actuator level. The fieldbus interconnects the devices at field level. Characteristic feature of the fieldbus is their highly reliable transfer of signals and real-time response.

Galvanic coupling

A galvanic coupling occurs when two circuits use the same cable. Typical sources of interference are, for example, starting motors, static discharges, clocked devices, and a potential difference between the housing of components and the common power supply.

GND

Abbreviation for “GROUND” (0 potential).

Ground

In electrical engineering the name for conductive grounding with an electrical potential at any point equal to zero. In the environment of grounding devices, the electrical ground potential may not equal zero. This is called a “reference ground”.

Ground (verb)

Represents the connection of an electrically conductive component to the equipotential earth via a grounding device.

Ground connection

One or several components that have a direct and good contact with the ground.

GSD

The device master data files (GSD) contain standardized PROFIBUS station descriptions. They are used to simplify the configuration of the DP master and DP slaves.

Hexadecimal

Number system with base 16. Counting from 0 to 9 and then with the letters A, B, C, D, E and F.

I/O
Abbreviation for “Input/Output”.

Impedance
Apparent resistance that a component or circuit of several components has for an alternating current at a particular frequency.

Inactive metal parts
Conductive parts that cannot be touched and which are insulated from active metal parts. They can, however, carry voltage in the event of a fault.

Inductive coupling
Inductive (magnetic) coupling occurs between two current carrying conductors. The magnetism produced by the currents induces an interference voltage. Typical interference sources are, for example transformers, motors, mains cables installed parallel and RF signal cables.

Lightning protection
Represents all measures for preventing system damage due to overvoltage caused by lightning strike.

Low impedance connection
Connection with low alternating-current resistance.

LSB
Abbreviation for “Least Significant Bit”. Bit with the least significant value.

Master
Station or node in a bus system that controls communication between the other stations of the bus system.

Master-slave mode
Operating mode in which a station or node of the system acts as master that controls communication on the bus.

Mode
Operating mode.

Module bus
Represents the internal bus of an XI/ON station. Used by the XI/ON modules for communication with the gateway. Independent of the fieldbus.

MSB
Abbreviation for “Most Significant Bit”. Bit with the most significant value.

Multimaster mode
Operating mode in which all stations or nodes of a system have equal rights for communicating on the bus.

NAMUR

Abbreviation for “Normen-Arbeitsgemeinschaft für Mess- und Regeltechnik” (Standards Work Group for Instruments and Controls). NAMUR proximity switches represent a special category of 2-wire proximity switches. They are highly resistant to interference and reliable due to their special construction, e.g. low internal resistance, few components and short design.

Overhead

System management time required in the system in each transmission cycle.

Parameter assignment

Assignment of parameters in the configuration software of the DP master for the individual stations on the bus and their modules.

PLC

Abbreviation for Programmable Logic Controller.

Potential-free

Galvanic isolation between the reference potentials of the control and load circuit of I/O modules.

PROFIBUS-DP

PROFIBUS bus system with the DP protocol. DP stands for “decentralized periphery”.

PROFIBUS-DP is based on DIN 19245 Part 1+4, and was integrated in the European fieldbus standard EN 50170. It is used for high-speed data exchange between the central DP master and the decentralized peripheral devices, the DP slaves. The comprehensive use is implemented by means of a multi-master concept.

PROFIBUS-DP address

Each PROFIBUS-DP station is assigned an unambiguous PROFIBUS-DP address by means of which it can be addressed by the master.

PROFIBUS-DP master

The PROFIBUS-DP master is the central station and controls the PROFIBUS access of all PROFIBUS-DP slaves.

PROFIBUS-DP slave

PROFIBUS-DP slaves are addressed by the PROFIBUS-DP master and exchange data with it at its request.

Protected against short-circuit

Property of electrical equipment. Short-circuit-proof equipment has the ability to withstand the thermal and dynamic loads that may occur at the location of installation on account of a short-circuit.

Protective conductor
A conductor required for the protection against dangerous currents, designated by the letters PE (abbreviation of “Protective Earth”).

Radiated coupling
Radiated coupling occurs when an electromagnetic wave makes contact with a conductor structure. The impact of the wave induces currents and voltages. Typical interference sources are, for example ignition circuits (spark plugs, commutators of electrical motors) and transmitters (e.g. radio-operated devices), which are operated near the corresponding conductor structure.

Reference ground
Ground potential in the area of grounding devices. Unlike "ground", which always has zero potential, it may have any potential except zero.

Reference potential
Represents a reference point for measuring and/or visualizing the voltage of any connected electrical circuits.

Repeater
Amplifier for signals transferred across a bus.

Response time
In a bus system the time interval between the sending of a read job and the receipt of the response. Within an input module, it represents the time interval between the signal change at an input and its output to the bus system.

RS 485
Serial interface in accordance with the EIA standard for high-speed data transmission via several transmitters.

Serial
Describes an information transfer technique. Data is transferred in a bit-stream across the cables.

Shield
Term that describes the conductive covering of cables, cubicles and cabinets.

Shielding
All measures and equipment used for connecting system parts with the shield.

Slave
Station in a bus system that is subordinate to the master.

Station
Function unit or module, consisting of several elements.

SUB-D plug
9-pole plug for connecting the fieldbus.

Topology

Geometric structure of a network or circuit arrangement.

UART

Abbreviation for “Universal Asynchronous Receiver/ Transmitter”. A UART is a logic circuit used for converting an asynchronous serial data sequence into a bit-parallel data sequence or vice versa.

Unidirectional

Working in one direction.

Index

Eaton Electrical
1000 Cherrington Parkway
Moon Township, PA 15108-4312
USA
Tel: 1-800-525-2000
www.EatonElectrical.com

EATON

Printed in USA
Publication No. MN05013001E
May 2005